

```

/
*****
*****
Name      : CaseAssignmentEngine_AC
Author    : @@CBIN0VI
Date      : 08/14/2012
@Description : Case Assignment Engine

Modified   : [Anjali] 22 Feb 2013 : #03260706, #03260620
Modified   : [Devaki] 4 March 2013: #03267942
Modified   : [Rahul] 3 May 2017: SFDC-4708
*****
*****/
@SuppressWarnings('PMD')
global class CaseAssignmentEngine_AC {

    public static final String CHECK_ASSIGNMENT = 'Check
Assignment';
    public static final String IN_DELAY = 'In Delay';
    public static final String IN_DELAY_CS_TEMP = 'In Delay
CS_Temp';
    public static final String DELAY_REASSIGN = 'Delay
Reassign';
    public static final String NO_ASSIGNMENT = 'No Assignment';
    public static final String ASSIGNED = 'Assigned';
    public static final String NO_CASE_OWNER_STATUS = 'No Case
Owner';
    public static final String NO_CASE_OWNER_REASSIGN_STATUS =
'No Case Owner Reassign';
    public static final String FIND_ASSIGNMENT = 'Find
Assignment';
    public static final String GCC_ALLEGRO_ASSIGNMENT =
'GCC_Allegro Assignment'; //Ticket: 03279500 Kalpesh, Mar 25, 2013
    public static final String ACTIVE = 'Active'; //SFDC-4707
changes for LUT query
    public static final String GCS_ASSIGNMENT = 'GCS
Assignment'; //SFDC-4709 changes
    public static final String NACS_ASSIGNMENT = 'NACS
Assignment'; //SFDC-4709 changes
    public static final String NACS = 'NACS'; //SFDC-4709
changes
    private static string NO_DELAY = 'nodelayrule';

    public static final String SSG_QUEUE = 'SSG'; //DELETE THIS
    public static final String SSG_CHINA = 'SSG_China';
    public static final String SSG_FRANCE = 'SSG_France';
    public static final String SSG_GERMANY = 'SSG_Germany';
    public static final String SSG_HONG_KONG = 'SSG_Hong_Kong';
    public static final String SSG_INDIA = 'SSG_India';
    public static final String SSG_ISREAL = 'SSG_Israel';
    public static final String SSG_JAPAN = 'SSG_Japan';
    public static final String SSG_KOREA = 'SSG_Korea';

```

```
        public static final String SSG_NORTH_AMERICA =
'SSG_North_America';
        public static final String SSG_RUSSIA = 'SSG_Russia';
        public static final String SSG_SINGAPORE = 'SSG_Singapore';
        public static final String SSG_TAIWAN = 'SSG_Taiwan';
        public static final String SSG_UNITED_KINGDOM =
'SSG_United_Kingdom';
        public static final String SSG_UNITED_STATES =
'SSG_United_States';
```

```
        public static final String EDA_SUPPORT = 'EDA_SUPPORT';
        public static final String HDS_APPLICATION =
'HDS_APPLICATION';
        public static final String HDS_Chamber_ACC =
'HDS_Chamber_ACC';
        public static final String HDS_CONNECT = 'HDS_CONNECT';
        public static final String HDS_TECHNOLOGY =
'HDS_TECHNOLOGY';
        public static final String SIGRITY_TEAM = 'SIGRITY_TEAM';
        public static final String ILS_EDU = 'ILS_EDU';
        public static final String GCC_ALLEGRO='GCC_Allegro'; //
```

Ticket: 03279500 Kalpesh, Mar 25, 2013

```
        private static final String GCS ='GCS';/**'Global-
Noida'*/ //Jyoti Goyal, JIRA issue SFDC-1595,28/4/15
```

```
        private static final String
DIP_ETHERNET_SUPPORT='DIP_ETHERNET_SUPPORT'; //Kalpesh, May 8,
Ticket: 03299280
```

```
        private static final String DOD = 'Department of
Defense'; //Kalpesh, July 24, 2014, ASR 639345
        private static final String ALL = 'ALL';//Kalpesh, Oct 29,
2015, SFDC 2089
```

```
        private static Id EDA_SUPPORTQueue;
        private static Id HDS_APPLICATIONQueue;
        private static Id HDS_Chamber_ACCQueue;
        private static Id HDS_CONNECTQueue;
        private static Id HDS_TECHNOLOGYQueue;
        private static Id SIGRITY_TEAMQueue;
        private static Id ILS_EDUQueue;
        private static Id GCC_AllegroQueue;//Ticket: 03279500
Kalpesh, Mar 25, 2013
```

```
        public static final String SUPPORT_CHINA = 'China';
        public static final String SUPPORT_FRANCE = 'France';
        public static final String SUPPORT_GERMANY = 'Germany';
        public static final String SUPPORT_HONG_KONG = 'Hong Kong';
        public static final String SUPPORT_INDIA = 'India';
        public static final String SUPPORT_ISREAL = 'Israel';
        public static final String SUPPORT_JAPAN = 'Japan';
        public static final String SUPPORT_KOREA = 'Korea';
        public static final String SUPPORT_NORTH_AMERICA = 'North
America';
        public static final String SUPPORT_RUSSIA = 'Russia';
        public static final String SUPPORT_SINGAPORE = 'Singapore';
```

```

        public static final String SUPPORT_TAIWAN = 'Taiwan';
        public static final String SUPPORT_UNITED_KINGDOM = 'United
Kingdom';
        public static final String SUPPORT_UNITED_STATES = 'United
States';
        private static Map<Id,List<Id>> itarQueueMemberMap;

        private static List<Id> groupMemberList = new List<Id>();
        private static List<Id> productFeatureList = new List<Id>();
        private static Set<Id> productFeatureListLevel1 = new
Set<Id>(); //Manish Shukla
        private static Set<Id> productFeatureListLevel2 = new
Set<Id>(); //Manish Shukla
        public static String itarcaseassign;
        public static Set<Id> itarApprovedUserIds = new Set<Id>();
        private static Set<Id> productClassList = new Set<Id>();//
Kalpesh May 7, 2013, ASR 565916
        private static Set<String> ASGIdSet = new Set<String>(); //
T-96472
        private static Map<String, Set<String>>
asgProdFeatIdsMap; // T-96472
        private static Map<Id, GroupMember> groupSSGChinaMemberList
= new Map<Id, GroupMember>();
        private static Map<Id, GroupMember> groupSSGFranceMemberList
= new Map<Id, GroupMember>();
        private static Map<Id, GroupMember>
groupSSGGermanyMemberList = new Map<Id, GroupMember>();
        private static Map<Id, GroupMember>
groupSSGHongKongMemberList = new Map<Id, GroupMember>();
        private static Map<Id, GroupMember> groupSSGIndiaMemberList
= new Map<Id, GroupMember>();
        private static Map<Id, GroupMember> groupSSGISrealMemberList
= new Map<Id, GroupMember>();
        private static Map<Id, GroupMember> groupSSGJapanMemberList
= new Map<Id, GroupMember>();
        private static Map<Id, GroupMember> groupSSGKoreaMemberList
= new Map<Id, GroupMember>();
        private static Map<Id, GroupMember>
groupSSGNorthAmericaMemberList = new Map<Id, GroupMember>();
        private static Map<Id, GroupMember> groupSSGRussiaMemberList
= new Map<Id, GroupMember>();
        private static Map<Id, GroupMember>
groupSSGSingaporeMemberList = new Map<Id, GroupMember>();
        private static Map<Id, GroupMember> groupSSGTaiwanMemberList
= new Map<Id, GroupMember>();
        private static Map<Id, GroupMember>
groupSSGUnitedKingdomMemberList = new Map<Id, GroupMember>();
        private static Map<Id, GroupMember>
groupSSGUnitedStatesMemberList = new Map<Id, GroupMember>();

        private static Map<Id, Product_Support_Agent__c>
groupNormalChinaMemberList = new Map<Id,
Product_Support_Agent__c>();
        private static Map<Id, Product_Support_Agent__c>

```

```

groupNormalFranceMemberList = new Map<Id,
Product_Support_Agent__c>();
    private static Map<Id, Product_Support_Agent__c>
groupNormalGermanyMemberList = new Map<Id,
Product_Support_Agent__c>();
    private static Map<Id, Product_Support_Agent__c>
groupNormalHongKongMemberList = new Map<Id,
Product_Support_Agent__c>();
    private static Map<Id, Product_Support_Agent__c>
groupNormalIndiaMemberList = new Map<Id,
Product_Support_Agent__c>();
    private static Map<Id, Product_Support_Agent__c>
groupNormalIsraelMemberList = new Map<Id,
Product_Support_Agent__c>();
    private static Map<Id, Product_Support_Agent__c>
groupNormalJapanMemberList = new Map<Id,
Product_Support_Agent__c>();
    private static Map<Id, Product_Support_Agent__c>
groupNormalKoreaMemberList = new Map<Id,
Product_Support_Agent__c>();
    private static Map<Id, Product_Support_Agent__c>
groupNormalNorthAmericaMemberList = new Map<Id,
Product_Support_Agent__c>();
    private static Map<Id, Product_Support_Agent__c>
groupNormalRussiaMemberList = new Map<Id,
Product_Support_Agent__c>();
    private static Map<Id, Product_Support_Agent__c>
groupNormalSingaporeMemberList = new Map<Id,
Product_Support_Agent__c>();
    private static Map<Id, Product_Support_Agent__c>
groupNormalTaiwanMemberList = new Map<Id,
Product_Support_Agent__c>();
    private static Map<Id, Product_Support_Agent__c>
groupNormalUnitedKingdomMemberList = new Map<Id,
Product_Support_Agent__c>();
    private static Map<Id, Product_Support_Agent__c>
groupNormalUnitedStatesMemberList = new Map<Id,
Product_Support_Agent__c>();
    private static Map<Id, Product_Support_Agent__c>
groupNormalGCCAllegroMemeberList = new Map<Id,
Product_Support_Agent__c>();
    private static Map<Id, Product_Support_Agent__c>
groupNormalGCSNAMemberList = new Map<Id,
Product_Support_Agent__c>(); //SFDC-4709 changes
    private static Map<Id, Product_Support_Agent__c>
groupNormalNACSMemberList = new Map<Id,
Product_Support_Agent__c>(); //SFDC-4709 changes

    //Kalpesh, Dec 30, 2016, SFDC-4205 - START
    //Group List for PFL1
    private static Map<Id, Product_Support_Agent__c>
groupNormalPFL1ChinaMemberList = new Map<Id,
Product_Support_Agent__c>();
    private static Map<Id, Product_Support_Agent__c>

```

```

groupNormalPFL1FranceMemberList = new Map<Id,
Product_Support_Agent__c>();
    private static Map<Id, Product_Support_Agent__c>
groupNormalPFL1GermanyMemberList = new Map<Id,
Product_Support_Agent__c>();
    private static Map<Id, Product_Support_Agent__c>
groupNormalPFL1HongKongMemberList = new Map<Id,
Product_Support_Agent__c>();
    private static Map<Id, Product_Support_Agent__c>
groupNormalPFL1IndiaMemberList = new Map<Id,
Product_Support_Agent__c>();
    private static Map<Id, Product_Support_Agent__c>
groupNormalPFL1IsrealMemberList = new Map<Id,
Product_Support_Agent__c>();
    private static Map<Id, Product_Support_Agent__c>
groupNormalPFL1JapanMemberList = new Map<Id,
Product_Support_Agent__c>();
    private static Map<Id, Product_Support_Agent__c>
groupNormalPFL1KoreaMemberList = new Map<Id,
Product_Support_Agent__c>();
    private static Map<Id, Product_Support_Agent__c>
groupNormalPFL1NorthAmericaMemberList = new Map<Id,
Product_Support_Agent__c>();
    private static Map<Id, Product_Support_Agent__c>
groupNormalPFL1RussiaMemberList = new Map<Id,
Product_Support_Agent__c>();
    private static Map<Id, Product_Support_Agent__c>
groupNormalPFL1SingaporeMemberList = new Map<Id,
Product_Support_Agent__c>();
    private static Map<Id, Product_Support_Agent__c>
groupNormalPFL1TaiwanMemberList = new Map<Id,
Product_Support_Agent__c>();
    private static Map<Id, Product_Support_Agent__c>
groupNormalPFL1UnitedKingdomMemberList = new Map<Id,
Product_Support_Agent__c>();
    private static Map<Id, Product_Support_Agent__c>
groupNormalPFL1UnitedStatesMemberList = new Map<Id,
Product_Support_Agent__c>();
    private static Map<Id, Product_Support_Agent__c>
groupNormalPFL1GCCAllegroMemeberList = new Map<Id,
Product_Support_Agent__c>();
    private static Map<Id, Product_Support_Agent__c>
groupNormalPFL1GCSNAMemberList = new Map<Id,
Product_Support_Agent__c>(); // #SFDC-4709 changes
    private static Map<Id, Product_Support_Agent__c>
groupNormalPFL1NACSMemberList = new Map<Id,
Product_Support_Agent__c>(); // #SFDC-4709 changes

    //Group List for PFL2
    //
    private static Map<Id, Product_Support_Agent__c>
groupNormalPFL2ChinaMemberList = new Map<Id,
Product_Support_Agent__c>();
    private static Map<Id, Product_Support_Agent__c>

```

```

groupNormalPFL2FranceMemberList = new Map<Id,
Product_Support_Agent__c>();
    private static Map<Id, Product_Support_Agent__c>
groupNormalPFL2GermanyMemberList = new Map<Id,
Product_Support_Agent__c>();
    private static Map<Id, Product_Support_Agent__c>
groupNormalPFL2HongKongMemberList = new Map<Id,
Product_Support_Agent__c>();
    private static Map<Id, Product_Support_Agent__c>
groupNormalPFL2IndiaMemberList = new Map<Id,
Product_Support_Agent__c>();
    private static Map<Id, Product_Support_Agent__c>
groupNormalPFL2IsraelMemberList = new Map<Id,
Product_Support_Agent__c>();
    private static Map<Id, Product_Support_Agent__c>
groupNormalPFL2JapanMemberList = new Map<Id,
Product_Support_Agent__c>();
    private static Map<Id, Product_Support_Agent__c>
groupNormalPFL2KoreaMemberList = new Map<Id,
Product_Support_Agent__c>();
    private static Map<Id, Product_Support_Agent__c>
groupNormalPFL2NorthAmericaMemberList = new Map<Id,
Product_Support_Agent__c>();
    private static Map<Id, Product_Support_Agent__c>
groupNormalPFL2RussiaMemberList = new Map<Id,
Product_Support_Agent__c>();
    private static Map<Id, Product_Support_Agent__c>
groupNormalPFL2SingaporeMemberList = new Map<Id,
Product_Support_Agent__c>();
    private static Map<Id, Product_Support_Agent__c>
groupNormalPFL2TaiwanMemberList = new Map<Id,
Product_Support_Agent__c>();
    private static Map<Id, Product_Support_Agent__c>
groupNormalPFL2UnitedKingdomMemberList = new Map<Id,
Product_Support_Agent__c>();
    private static Map<Id, Product_Support_Agent__c>
groupNormalPFL2UnitedStatesMemberList = new Map<Id,
Product_Support_Agent__c>();
    private static Map<Id, Product_Support_Agent__c>
groupNormalPFL2GCCAllegroMemeberList = new Map<Id,
Product_Support_Agent__c>();
    private static Map<Id, Product_Support_Agent__c>
groupNormalPFL2GCSNAMemberList = new Map<Id,
Product_Support_Agent__c>(); //SFDC-4709 changes
    private static Map<Id, Product_Support_Agent__c>
groupNormalPFL2NACSMemberList = new Map<Id,
Product_Support_Agent__c>(); //SFDC-4709 changes
    //Kalpesh, Dec 30, 2016, SFDC-4205 - END
    //
    private static Id SSGChinaQueue;
    private static Id SSGFranceQueue;
    private static Id SSGGermanyQueue;
    private static Id SSGHongKongQueue;
    private static Id SSGIndiaQueue;

```

```

        private static Id SSGIsrealQueue;
        private static Id SSGJapanQueue;
        private static Id SSGKoreaQueue;
        private static Id SSGNorthAmericaQueue;
        private static Id SSGRussiaQueue;
        private static Id SSGSignaporeQueue;
        private static Id SSGTaiwanQueue;
        private static Id SSGUnitedKingdomQueue;
        private static Id SSGUnitedStatesQueue;
        private static Id CVASupportUserId;
        private static Id VIPSupportUserId;//Kalpesh, Feb 21, 2013
Ticket#03263166
        private static final String PALLADIUM_XP ='Palladium
XP' ; //Kalpesh Feb 13
        //private static final String DEFAULT_CASE_QUEUE = 'Default
case Queue';
        public static final String DELAY_QUEUE = 'CS_Delay';
        private static final String GCC_ICD_STD = 'GCC_ICD_STD';
        public static final String CS_TEMP = 'CS_TEMP';
        //private static final String E2E_Pool = 'E2E Pool';
        private static final String DIP = 'DIP_SUPPORT';
        private static final String VIP = 'VIP Support';
        private static final String NO_CASE_OWNER = 'NO_CASE_OWNER';
        private static final String EMAIL_SOURCE = 'Email';
        private static final String CVA_SUPPORT='CVA_Support'; //
Kalpesh Feb 16, #432
        private static Id SSGQueueId;
        //private static Id defaultQueue;
        private static Id delayQueue;
        private static Id GCC_ICD_STDQueue;
        private static Id CS_TEMPQueue;
        //private static Id E2E_PoolQueue;
        private static Id DIPQueue;
        private static Id NO_CASE_OWNERQueue;
        private static Map<String, String> ASGToProductFeatureMap =
new Map<String,String>();
        private static Map<String, String> advancedSupportGroupMap =
new Map<String,String>();
        private static Map<String, Group> groupMap = new Map<String,
Group>();
        private static Map<String, String> contactASGMap = new
Map<String,String>();
        private static Map<String, List<Case>> ASGToCaseMap = new
Map<String,List<Case>>();
        private static Support_Center_Configuration__c configRec =
new Support_Center_Configuration__c() ;

        private static List<ASG_to_Prod_Feat_Junc__c>
asgProdFeatList;
        public static map<Id, Product_Support_Agent__c>
supportAgentMap;
        private static map<Id, map<id, String>>
supportAgentExpertiseMap;
        private static map<Id, integer> scoreMap;

```

```

        //private static Map<String, String>
groupMemberSupportCenterMap = new Map<String, String>();
        private static Map<String, DateTime> groupMemberMap = new
Map<String, DateTime>();
        public static boolean alreadyRan = false;
        private static Map<String, String> caseProductQueues = new
Map<String, String>();

        private static List<Case> asgCaseChatter = new List<Case>();
        public static Map<Integer, Case> asgCaseChatterMap = new
Map<Integer, Case>(); //Anjali : #03260706 #03260620

        // private static List<Queued_Task__c> queuedTaskList = new
List<Queued_Task__c>(); SFDC-13490 Code Committed

        // SFDC-13490 Code Added
        private static List<Task> queuedTaskList = new List<Task>();

        private static String productClassNameStr;
        private static Set<String> ownerSet;
        private static Map<String , String> ownerMap;
        private static List<Case> caseToUpdateManager ;
        private static Boolean isNormalAssignmentFailed;

        public static String SIGRITY_PRODUCT = 'Sigrity Power and
Signal Integrity';

        private static Set<String> gccAllegroStatus = new
Set<String> {'New', 'Transfer'};
        private static Set<String> productFeatureForTierAcccount =
new Set<String> { // Devaki : #03267942
                'Allegro Design Entry CIS', 'AMS
Simulator' , 'OrCAD Layout', 'OrCAD PCB Editor'};

        private static Set<String> productFeaturesDDRForDIP = new
Set<String>{ //Kalpesh asr 566496, Mar 22, 2013
                'DDR PHY', 'DDR Controller' };
        private static boolean byPassDDRProdFeatCheck = false;
        //ASR 568432, Kalpesh Vyas, Jan 07, 2013
        public static Id GCCAllegroId, TempQueueId, IPGQueueId; //ASR
566413, Kalpesh Vyas, Sep 17, 2013,
        public static Id RadarElbaQueueId, RadarMystQueueId; // JIRA
Issue SFDC-1408 || Prateek
        public static String TEMP_QUEUE='NO_CASE_OWNER_TEMP'; //ASR
566413, Kalpesh Vyas, Sep 17, 2013
        //ASR 567602, Kalpesh August 16, 2013
        private static Integer CS_DELAY_TIME = 18;
        private static Integer CS_TEMP_DELAY_TIME = 1;
        //ASR 567878, Kalpesh Oct 23, 2013
        private static Set<String> byPassCSDelay_BUSummaries = new
Set<String>();
        private static String byPassCSDelayReason = '';
        //ASR 568432, Kalpesh Vyas, Jan 07, 2013 START

```



```

        private static string IPG_QUEUE='IPG_SUPPORT';
        private static string Radar_Myst_Queue =
'Radar_Myst_Queue'; // JIRA Issue SFDC-1408 || Prateek
        private static string Radar_Elba_Queue =
'Radar_Elba_Queue'; // JIRA Issue SFDC-1408 || Prateek
        private static string OTHER='Other/Other';
        //ASR 568432, Kalpesh Vyas, Jan 07, 2013 END
        private static Map<Id, Map<Id, String>>
prodFeatLevel1UserExpertiseMap;
        private static Map<Id, Map<Id, String>>
prodFeatLevel2UserExpertiseMap;
        private static Map<Id, Map<Id, Set<String>>>
prodFeatLevel1UserSupportCenters;
        private static Map<Id, Map<Id, Set<String>>>
prodFeatLevel2UserSupportCenters;
        public static Map<String, Lookup_Table__c>
LUTSupLevelMap; //SFDC-4707 changes
        public static Map<String, Lookup_Table__c>
LUTEcomSupLevelMap;
        private static Map<Id, Boolean> dipWithOverrideReasonMap =
new Map<Id, Boolean>();
        private static Map<String, String> contactCopaIDMap = new
Map<String, String>();
        private static Map<String, IP_Config__c> copaIdIpConfigMap =
new Map<String, IP_Config__c>();
        public static Map<String, String> dipProjectQueueMap = new
Map<String, String>();
        //ASR 567842, Kalpesh Nov 07, 2013 -- START
        private static Set<String> byPassLicenceProdStatus;

/**
 * @description Static variable that sets the bypass license
status and return the value of the same
 */
        public static Set<String> byPassLicenceProductStatus {
            get{
                if(byPassLicenceProdStatus != null &&
byPassLicenceProdStatus.size()>0)
                    return byPassLicenceProdStatus;
                try{
                    CaseAssignmentRelatedValues__c
prodStatusByPassDelay =
CaseAssignmentRelatedValues__c.getInstance('ProdStatusByPassDelay');
                    byPassLicenceProdStatus = new Set<String>();
                    for(String
s :prodStatusByPassDelay.ValueList__c.split('\n',0)){
byPassLicenceProdStatus.add(s.trim());
                    }
                }
                Catch(Exception ex){
                    byPassCSDelay_BUSummaries = new
Set<String>{'Eval'};

```

```

    }
    return byPassLicenceProdStatus;
}
}
//ASR 567842, Kalpesh Nov 07, 2013 --END

static{//Ticket: 03279500 Kalpesh, Mar 25, 2013
    if(GCCAllegroId == null){
        //ASR 566413, Kalpesh Vyas, Sep 17, 2013
        for(Group g:[Select Id, Name FROM Group
Where
                                Name IN
(:GCC_ALLEGRO, :TEMP_QUEUE, :IPG_QUEUE, :Radar_Myst_Queue, :Radar_Elba_Queue) Limit 10]){
GCCAllegroId = g.Id;
                                if(g.Name == GCC_ALLEGRO)
                                if(g.Name == TEMP_QUEUE) TempQueueId
= g.Id;
                                if(g.Name == IPG_QUEUE) IPGQueueId =
g.Id;
                                if(g.Name == Radar_Myst_Queue)
RadarMystQueueId = g.Id; // JIRA Issue SFDC-1408 || Prateek
                                if(g.Name == Radar_Elba_Queue)
RadarElbaQueueId = g.Id; // JIRA Issue SFDC-1408 || Prateek
                                }
        }
        //ASR 567878, Kalpesh Oct 23, 2013
        try{
            CaseAssignmentRelatedValues__c
            buSummaryBypassDelay =
            CaseAssignmentRelatedValues__c.getInstance('BUSummaryBypassDelay');
            for(String
            s :buSummaryBypassDelay.ValueList__c.split('\n',0)){
            byPassCSDelay_BUSummaries.add(s.trim());
            byPassCSDelayReason += ', '+
            s.trim();
            }
            if(byPassCSDelayReason != '')
            byPassCSDelayReason =
            byPassCSDelayReason.substring(2);

            }Catch(Exception ex){
                byPassCSDelay_BUSummaries = new
                Set<String>{'Silicon Package Board'};
            }

            /* #SFDC-4707 Get Lookup Table Support Level records
Start*/
            LUTSupLevelMap = new Map<String, Lookup_Table__c>();
            LUTEcomSupLevelMap = new Map<String,
Lookup_Table__c>();
            for(Lookup_Table__c lut:[SELECT

```

```

Id,Support_Center__c,Parent_Account__c,Product_Platform__c,Support_Level__c,Status__c,recordType.developername,Contact_Support_Level__c,

Tier_Support__c,Entitlement_Product_Status__c,Entitlement_Maintenance_Status__c,Product_Class__c from Lookup_Table__c where
Status__c=:ACTIVE and Support_Level__c!=null])
    {
        /* Assign the Support level value to map
with Support Center,Parent Account and Product Platform as a key
value */
        if(lut.recordType.developername == null ||
lut.recordType.developername == '' || lut.recordType.developername
== 'Basic') {

LUTSupLevelMap.put(lut.Support_Center__c+lut.Parent_Account__c+lut.P
roduct_Platform__c,lut);
        }
        else if(lut.recordType.developername ==
'Ecommerce') {
            if(lut.Product_Class__c == null)

LUTEcomSupLevelMap.put(lut.Contact_Support_Level__c+lut.Tier_Support
__c+lut.Entitlement_Product_Status__c+lut.Entitlement_Maintenance_St
atus__c,lut);
            else

LUTEcomSupLevelMap.put(lut.Contact_Support_Level__c+lut.Tier_Support
__c+lut.Entitlement_Product_Status__c+lut.Entitlement_Maintenance_St
atus__c+lut.Product_Class__c,lut);
        }
    } //For ends
    /* #SFDC-4707 Get Lookup Table Support Level records
End*/

    /* DIP Case Enterprise Project Assignment */
    for(Enterprise_Project_Assignment__mdt enProj:
[SELECT Id, Source__c, Queue_User_Id__c from
Enterprise_Project_Assignment__mdt where Source__c!=null and
Queue_User_Id__c!=null]){

dipProjectQueueMap.put(enProj.Source__c,enProj.Queue_User_Id__c);
    }

}

//ASR 568432, Kalpesh Vyas, Jan 08, 2013
public static Map<Id, Boolean>
getValidEntitlementCheckForDIPCases(List<Case> csList){

    Map<Id, Boolean> caseValidEntitlementMap = new Map<Id,
Boolean>();
    Map<String, Set<String>> contactProjectMap = new
Map<String, Set<String>>();

```

```

        Map<Id, List<IP_Config__c>> contactIPConfigMap = new
Map<Id, List<IP_Config__c>>();
        Map<Id, List<Case>> contactCaseMap = new Map<Id,
List<Case>>();
        list<String> caseEntProjectNames = new list<String>();
        set<String> caseProductFeatureNames = new set<String>();
        try{
            for(Case cs : csList){
                if(!
contactCaseMap.containsKey(cs.ContactId)){
                    contactCaseMap.put(cs.ContactId, new
List<Case>());
                }
                contactCaseMap.get(cs.ContactId).add(cs);

                if(cs.Enterprise_Project_Name__c!
=null)caseEntProjectNames.add(cs.Enterprise_Project_Name__c);

                if(cs.Product_Feature2__c != null)
caseProductFeatureNames.add(cs.Product_Feature2__c);
            }

            for(EntitlementContact ec :[Select EntitlementId,
ContactId, Entitlement.Status__c,

            Entitlement.Parent_Entitlement__c,
Entitlement.COPA_Integration_Id__c

            From EntitlementContact Where ContactId
IN :contactCaseMap.keySet()

            and Entitlement.Status__c = 'Active'

            and (not (Entitlement.Locking_Id__c like '%TIPP' OR
Entitlement.Locking_Id__c like '%TIP' )))]{//COSII-4390, not to
consider TIP entitlements

                if( ec.Entitlement.COPA_Integration_Id__c !=
null

                    //&&
ec.Entitlement.Parent_Entitlement__c == null //Considering only
parent entitlements for now

                    ){
                        if(!
contactProjectMap.containsKey(ec.Entitlement.COPA_Integration_Id__c)
){

                            contactProjectMap.put(ec.Entitlement.COPA_Integration_Id__c, new
Set<String>());

                        }

                        contactProjectMap.get(ec.Entitlement.COPA_Integration_Id__c).add(ec.
ContactId);

```

```

        }
    }

    if(contactProjectMap.size() > 0){
        for (IP_Config__c config :[Select id,
Databahn_Project__c, Maintenance_End_Date__c,
COPA_Integration_Id__c, Source__c, Product_Feature__c

        from IP_Config__c where COPA_Integration_Id__c in :
contactProjectMap.keySet() AND Product_Feature__c
IN :caseProductFeatureNames AND Maintenance_End_Date__c
>= :Date.today() ]){
            for(String conId :
contactProjectMap.get(config.COPA_Integration_Id__c)){
                if(!
contactIPConfigMap.containsKey(conId)){
                    contactIPConfigMap.put(conId, new List<IP_Config__c>());
                }

contactIPConfigMap.get(conId).add(config); // adding only valid
entitlements

if(caseEntProjectNames.contains(config.Databahn_Project__c) &&
config.Source__c!=null){

contactCopaIDMap.put(conId,config.COPA_Integration_Id__c);

copaIdIpConfigMap.put(config.COPA_Integration_Id__c,config);

            }
        }
    }

    system.debug('contactIPConfigMap ::'+contactIPConfigMap);
    for(Id caseContId : contactCaseMap.keySet()){
        List<Case> cases =
contactCaseMap.get(caseContId);
        for(Case cs : cases){
            caseValidEntitlementMap.put(cs.Id,
false);
            if(contactIPConfigMap.size()>0 &&
contactIPConfigMap.containsKey(caseContId)
&&
contactIPConfigMap.get(caseContId).size()>0){

if((cs.Origin.contains(EMAIL_SOURCE)) || cs.Databahn_Project__c ==
null
||
(cs.Databahn_Project__c!= null && cs.Enterprise_Project_Name__c ==
OTHER)){

```

```

caseValidEntitlementMap.put(cs.Id, true);
        }else{
            for(IP_Config__c
ipConfig : contactIPConfigMap.get(caseContId)){
if(cs.Databahn_Project__c == ipConfig.Id && cs.Product_Feature2__c
== ipConfig.Product_Feature__c){
caseValidEntitlementMap.put(cs.Id, true);
                }
            }
        }
    }
} catch(exception exc){
    system.debug('**** EXCEPTION ****'+exc);
Exception_Handler.catchException(exc,NULL,NULL,'getValidEntitlementC
heckForDIPCases','Case');//Exception handling
    }
    return caseValidEntitlementMap;
}

/**
 * @description Kalpesh, ASR 566456, July 23, 2014 -- seperated
the logic to be used from external class
 * @param caseRec The case record for which details needs to be
filled
 * @param caseWebServiceList
 * @return String Return the message
 */
public static String populateCaseRelatedFields(Case caseRec,
List<Case> caseWebServiceList){
    String msgOnPage = '';

    if(caseRec.Contact != null ) {
        caseRec.Assignment_Status__c =
CaseAssignmentEngine_AC.CHECK_ASSIGNMENT ;
        caseRec.Assignment_Reason__c += ' Case
Reassigned Manually ';
        // T-104361 - Noopur
        caseRec.Account_Name__c =
caseRec.Contact.Account.Name; //Kalpesh, ASR 639345, July 23, 2014
        caseRec.Account_Type__c =
caseRec.Contact.Account.Type;//Kalpesh, ASR 639345, July 23, 2014

        caseRec.Contact_Name__c =
caseRec.Contact.FirstName + ' ' + caseRec.Contact.LastName;
        caseRec.Support_Center__c =
caseRec.Contact.Support_Center__c;
        if(!caseRec.Ecommerce Case__c) {

```

```
caseRec.Support_Level__c =  
caseRec.Contact.Support_Level__c; //Kalpesh Vyas, ASR 568061, Nov 26,  
2013
```

```
caseRec.Advanced_Support_All_Products__c =  
caseRec.Contact.Advanced_Support_Group__r.All_Products__c;
```

```
caseRec.Advanced_Support_ChatterGroup__c =  
caseRec.Contact.Advanced_Support_Group__r.Chatter_Group__c;  
caseRec.Advanced_Support_CSC__c =  
caseRec.Contact.Advanced_Support_Group__r.CSC__c;  
caseRec.Advanced_Support_Flag1__c =  
caseRec.Contact.Advanced_Support_Group__r.Active__c;  
caseRec.Advanced_Support_Group_ID__c  
= caseRec.Contact.Advanced_Support_Group__r.Id;  
caseRec.Advanced_Support_Queue__c =  
caseRec.Contact.Advanced_Support_Group__r.Queue__c;  
caseRec.Contact_DBE_Flag__c =  
caseRec.Contact.DBE_Flag__c;  
}
```

```
caseRec.BU_Summary_Attribute__c =  
caseRec.Product__r.BU_Summary__c;  
caseRec.Product_Class_Name__c =  
caseRec.Product__r.Name;  
caseRec.Product_Feature_Queue__c =  
caseRec.Product_Feature__r.Queue__c;  
caseRec.Product_Feature_Name__c =  
caseRec.Product_Feature__r.Name;  
caseRec.Product_Feature_Level_1_name__c =  
caseRec.Product_Feature_Lvl_1__r.name; //Manish Shukla  
caseRec.Product_Feature_Level_2_name__c =  
caseRec.Product_Feature_Lvl_2__r.name; //Manish Shukla
```

```
system.debug('^^^^^^'+caseRec.Product_Feature_Level_1_name__c);
```

```
system.debug('^^^^^^'+caseRec.Product_Feature_Level_2_name__c);  
caseRec.Contact_Employee_Flag__c =  
caseRec.Contact.Employee_Flag__c; // Rashmi ASR :640278 Dec 15,2014  
caseRec.Parent_Account__c =  
caseRec.Contact.Parent_Account__c; //Kalpesh, ASR 639345, July 23,  
2014
```

```
//caseRec.Product_Feature_Name__c =  
caseRec.Product_Feature2__r.Name; //Kalpesh asr 566496 Mar 22, 2013  
if(caseWebServiceList != null)  
caseWebServiceList.add(caseRec);
```

```
System.debug('caseRec.Product_Feature2__r.Name  
' + caseRec.Product_Feature2__r.Name +  
caseRec.Product_Feature__r.Name  
' + caseRec.Product_Feature__r.Name);  
}
```

```
/*Jitendra-Oct-23/2013: ASR-567868; Case  
Reassignment should work for DIP even when contact is null*/
```

```

else if(caseRec.Product_Class_Name__c ==
'Design IP'){

    caseRec.Assignment_Status__c =
CaseAssignmentEngine_AC.CHECK_ASSIGNMENT ;
    caseRec.Assignment_Reason__c += ' Case
Reassigned Manually ';
    caseRec.BU_Summary_Attribute__c =
caseRec.Product__r.BU_Summary__c;
    caseRec.Product_Class_Name__c =
caseRec.Product__r.Name;
    caseRec.Product_Feature_Queue__c =
caseRec.Product_Feature__r.Queue__c;
    caseRec.Product_Feature_Name__c =
caseRec.Product_Feature__r.Name;
    caseRec.Product_Feature_Level_1_name__c =
caseRec.Product_Feature_Lvl_1__r.name; // Manish Shukla
    caseRec.Product_Feature_Level_2_name__c =
caseRec.Product_Feature_Lvl_2__r.name; // Manish Shukla
    if(caseWebServiceList != null)
        caseWebServiceList.add(caseRec);
    }
    else {
        msgOnPage = 'The product feature
selected does not have valid entitlement. '+
        'Please
either select a product feature with a valid' +
        ,
entitlement or indicate an override reason.';

    }
    return msgOnPage;
}
/
*****
***
    @description Method to call from button for case
reassignment from the button Reassign. Its a remote method call
    @param caseId Case ID to reassign
    @return String Success of failure message

*****
***/
    webService static String caseReassign (String caseId){

        String msgOnPage = '';
        List<String> caseIds = new List<String>();
        caseIds.add(caseId);
        Set<Id> rcoIds=new Set<Id>(); //Rashi,ASR 569633,
June 20 2014

```



```

List<Case> caseWebServiceList = new List<Case>();

Case csToUpdateCount;
for (Case caseRec: getCaseList(caseIds)) {
    //SFDC-3244, Kalpesh Vyas, May 17, 2016 START
    if(caseRec.Reassign_Click_count__c == null)
        caseRec.Reassign_Click_count__c = 0;

    caseRec.Reassign_Click_count__c += 1;
    CaseStaticData.isCDNLastModifiedUpdate = true; //
SFDC-3837: Gaurav, Do not update CDN lastmodified date / by for
reassignment of owner.
    csToUpdateCount = caseRec;
    //SFDC-3244, Kalpesh Vyas, May 17, 2016 END
    msgOnPage= populateCaseRelatedFields(caseRec,
caseWebServiceList);
}
if(caseWebServiceList.size() > 0) {
    byPassDDRProdFeatCheck = true;    //Kalpesh
asr 566496 Mar 22, 2013
    assignMultipleCases(caseWebServiceList,
null , false);
    //Rashi,ASR 569633, June 20 2014 --Start
    for(Restricted_Case_Owners__c
rco:Restricted_Case_Owners__c.getAll().values()){
        rcoIds.add(rco.Id__c);
    }
    for(Case caseRec:caseWebServiceList){
        if(caseRec.Status!= 'New'){
            if(caseRec.ownerId!=null &&
rcoIds.contains(caseRec.ownerId)){
                msgOnPage='Case in status other
than new can not be assigned to queue';
                return msgOnPage;
            }
        }
    }
    //Rashi,ASR 569633, June 20 2014 --END
    update caseWebServiceList;

CaseService.createAssignmentHistory(caseWebServiceList);

    }else if(msgOnPage == '' && csToUpdateCount!= null)
{ //SFDC-3244, Kalpesh Vyas, May 17, 2016
    update csToUpdateCount;
}
return msgOnPage;

}

/

*****
***

```

```

        @description Method for Batch
        @param caseToAssign of cases to assign
        @param delay
        @return List<Case>    List of cases objects with assignment
details
****/
    public static List<Case> assignBatchCases(List<Case>
caseToAssign, Boolean delay){
        List<String> caseIds = new List<String>();
        for (Case caseRec: caseToAssign) {
            caseIds.add(CaseRec.Id);
        }

        List<Case> caseListTemp = getCaseList(caseIds);
        //ASR 568510, Kalpesh, Mar 20, 2014
        Map<Id, Case> caseMaps = new Map<Id, Case>();

        for (Case caseRec: caseListTemp) {
            if (delay) {
                CaseRec.Assignment_Status__c =
CaseAssignmentEngine_AC.DELAY_REASSIGN;
            }
            else if(CaseRec.OwnerId == null ||
CaseRec.OwnerId !=GCCAllegroId){//Ticket: 03279500 Kalpesh, Mar 25,
2013
                CaseRec.Assignment_Status__c =
CaseAssignmentEngine_AC.NO_CASE_OWNER_REASSIGN_STATUS;
            }
            //ASR 568510, Kalpesh, Mar 20, 2014
            caseMaps.put(caseRec.Id, caseRec);
        }

        //ASR 568510, Kalpesh, Mar 20, 2014 --START
        for(Case newCase : [Select Id, OwnerId From Case
where id in :caseMaps.keySet()]){
            if(newCase.OwnerId !=
caseMaps.get(newCase.Id).OwnerId){
                caseMaps.remove(newCase.Id);
            }
        }

        assignMultipleCases(caseMaps.values(), null ,
false);

        return caseMaps.values();
        //ASR 568510, Kalpesh, Mar 20, 2014 --END
    }

/
****/

```

\*\*\*

@description Get Case Info  
@param caseIds Fetch the list of cases  
@return List<Case> return the list of cases objects with details

\*\*\*\*\*

\*\*\*/  
public static List<Case> getCaseList(List<Id> caseIds) {

//ASR 639345, Kalpesh, July 24, 2014  
//changed Account.Name and Account.Type to fetch  
from Contact  
List<Case> caseListTemp = [SELECT  
ITARCase\_\_c, AE\_Contact\_\_c,  
AE\_Contact\_Phone\_Number\_\_c, AccountId,  
Account\_Type\_\_c,  
Region\_While\_Active\_\_c, Alt\_Case\_Num\_\_c,  
Ecommerce\_Case\_\_c,  
Assigned\_Date\_\_c, Assignment\_Status\_\_c, //  
Assignment\_Reason\_\_c,  
BU\_Summary\_Attribute\_\_c, CC\_Emails\_\_c,  
CCR\_\_c, Id, CaseNumber, Origin,  
Type, IsClosed, Closed\_Class\_\_c, ClosedDate,  
ContactId, Contact.Ecommerce\_Entitlement\_\_c,  
Contact.Pure\_Ecommerce\_Customer\_\_c,  
Contact.Customer\_Type\_\_c,  
CreatedById, CreatedDate, IsDeleted,  
Description\_\_c, E2E\_\_c, Contact\_Employee\_Flag\_\_c, // Rashi ASR  
:640278 Dec 15,2014  
Engineering\_Owner\_\_c, EntitlementId,  
isEscalate\_\_c, Escalation\_Reason\_\_c,  
LastModifiedById, LastModifiedDate,  
MMAV\_Version\_\_c, Manager\_\_c,  
isMigrated\_Record\_\_c, SuppliedName,  
Override\_Reason\_\_c, OwnerId, Region\_\_c, Parent\_Account\_\_c, ParentId,  
Product\_\_c,  
Product\_Feature\_\_c, Product\_Feature2\_\_c,  
Product\_Feature\_Lvl\_1\_\_c, Product\_Feature\_Lvl\_2\_\_c,  
Product\_Feature\_Lvl\_2\_\_r.name,  
Product\_Feature\_Lvl\_1\_\_r.name, //Kalpesh Vyas, Feb 22, 2016  
Product\_Feature\_Version\_\_c, ProductId,  
RecordTypeId, isLicenseDelayAvoid\_\_c, //ASR 567842, Kalpesh Nov 07,  
2013  
Row\_Id\_\_c, Case\_Type\_\_c, Priority,  
State\_\_c, Status, Sub\_Class\_\_c, Subject, Support\_Level\_\_c,  
SystemModstamp, Test\_Delete\_Verfied\_\_c,  
Thread\_Id\_\_c, isFollowCase\_\_c, isTest\_\_c, Entitlement\_Status\_\_c,  
Assignment\_Reason\_\_c,  
Advanced\_Support\_Flag1\_\_c,  
Advanced\_Support\_All\_Products\_\_c, Advanced\_Support\_Queue\_\_c,  
Advanced\_Support\_Group\_ID\_\_c,  
Product\_Feature\_Queue\_\_c, Support\_Center\_\_c, Account\_Name\_\_c,  
Product\_Class\_Name\_\_c,  
Product\_Feature\_Name\_\_c, Assignment\_Made\_\_c,

```

Advanced_Support_ChatterGroup__c,
Contact_Name__c, To_Email_Address__c, No_Entitlement__c,
// T-104361 - Noopur
Contact.Account.Name, Contact.FirstName,
Contact.LastName, Contact.Support_Center__c,
Account.Support_Level__c,
Contact.DBE_Flag__c,
Product__r.BU_Summary__c, Product__r.Name,
Product_Feature__r.Queue__c,
Product_Feature__r.Name,
Contact.Advanced_Support_Group__r.All_Products__c,

Contact.Advanced_Support_Group__r.Chatter_Group__c,
Contact.Advanced_Support_Group__r.CSC__c,
Contact.Advanced_Support_Group__r.Active__c,
Contact.Advanced_Support_Group__r.Id,
Contact.Advanced_Support_Group__r.Queue__c,
Contact.Account.Type, Reassign_Click_count__c, //SFDC-3244
Contact.Support_Level__c,
Contact.Employee_Flag__c, Contact.Account_Type__c,
Contact.Case_Assignment__c, Contact.AE__c,
Contact.Case_Assignment__r.Name, Contact.Account.AE__c,
Contact.Parent_Account__r.AE__c, Account.AE__c,
Parent_Account__r.AE__c, //SFDC-3225, Kalpesh, Apr 01, 2016
Enterprise_Project_Name__c, TarBalls__c,
TarBalls__r.Name, //ASR 568432, Kalpesh Vyas, Jan 31, 2014
Databahn_Project__c,
Databahn_Project__r.Databahn_Project__c, //ASR 568432, Kalpesh Vyas,
Jan 07, 2014 */
Contact.Parent_Account__c,
Contact.Parent_Account__r.Name, //ASR 639345, Kalpesh, July 24, 2014
extCustomer_Project_Name__c, // JIRA Issue
SFDC-1408 || Prateek

Product__r.Product_Attribute__r.Cadence_Platform__c, // JIRA
#SFDC-4707
Product_Sold_By_Account__c, /* Project
Mainstream ::: Abhishek ::: */
Contact.Parent_Account_Type__c // JIRA
#SFDC-5350

FROM Case WHERE ID IN :caseIds];
for(case cs:caseListTemp){
    cs.Assignment_Reason__c = '';
    if(cs.ContactId!
=null)CaseStaticData.contParAccountType.put(cs.ContactId,cs.Contact.
Parent_Account_Type__c); // #SFDC-6132
}
return caseListTemp;

}

/**
@description Method for multiple Cases – called from Cases
@param caseToAssign CASEs for assignment

```

```
@param ASGProdFeatMap Advanced support group details  
@param isASGJctQueryRun  
*/  
public static void assignMultipleCases(List<Case>  
caseToAssign, Map<String, Set<String>> ASGProdFeatMap , boolean  
isASGJctQueryRun){  
  
ownerSet = new Set<String>();  
caseToUpdateManager = new  
List<Case>();  
  
//Workflows causing it to run twice;  
if (alreadyRan == false) {  
  
    alreadyRan = true;  
  
    System.debug('CASE: ' + caseToAssign);  
  
    //Get Config values  
for(Support_Center_Configuration__c sc :  
[SELECT Critical_Expert__c, Critical_Intermediate__c,  
Critical_Novice__c,  
Important_Expert__c, Important_Intermediate__c, Important_Novice__c,  
Minor_Expert__c,  
Minor_Intermediate__c,  
Minor_Novice__c, New__c, Transfer__c, Wait_CCR__c,  
Wait_Critical_CCR__c,  
Wait_Cust__c, Wait_Expert__c, Wait_Release__c, Working__c,  
Delay_Active__c  
FROM  
Support_Center_Configuration__c  
WHERE Active__c = true ORDER  
BY LastModifiedDate DESC LIMIT 1]){  
        configRec = sc;  
        }  
  
productFeatureList = new List<Id>();  
productClassList = new Set<Id>();//Kalpesh  
May 7, 2013, ASR 565916  
ASGIdSet = new Set<String>();  
  
//ASR 568432, Kalpesh Vyas, Jan 09, 2013 --  
START  
List<Case> casesForIPG = new List<Case>();  
List<Case> casesForRadar = new  
List<Case>(); // JIRA Issue SFDC-1408 || Prateek  
List<Case> tensilicaCases = new  
List<Case>(); // JIRA Issue SFDC-3225, Kalpesh, Apr 04, 2016  
Map<Id, Case> csCheckIPGMap = new Map<Id,  
Case>();  
for(Case cs : caseToAssign){  
cs.Assignment_Reason__c = '';  
if(cs.Product_Class_Name__c ==  
'Design IP'){  
  
if(cs.Origin == 'Radar'){ //
```

JIRA Issue SFDC-1408 || Prateek

```
                casesForRadar.add(cs);
            }
            else if(cs.ContactId ==
null){ // Case created through email will have project Other
                casesForIPG.add(cs);
            }else{

csCheckIPGMap.put(cs.Id, cs);

            }
            }else if(cs.Product_Class_Name__c ==
'Tensilica IP' || cs.Product_Class_Name__c == 'Tensilica SweRV'){//
Kalpesh, SFDC-3225, Apr 04, 2016
                tensilicaCases.add(cs);
            }
        }
    }
```

```
        // JIRA Issue SFDC-1408 || Prateek || START
        if(casesForRadar.size()>0){
            for(Case cs : casesForRadar){

if(cs.extCustomer_Project_Name__c == 'elba') {
                cs.ownerId =
RadarElbaQueueId;
                cs.Assignment_Reason__c +=
'Radar elba Assignment';
                cs.Assignment_Status__c =
ASSIGNED;
            }
            else
if(cs.extCustomer_Project_Name__c == 'myst'){
                cs.ownerId =
RadarMystQueueId;
                cs.Assignment_Reason__c +=
'Radar myst Assignment';
                cs.Assignment_Status__c =
ASSIGNED;
            }
        }
    }
```

```
System.debug('tensilicaCases ::'+tensilicaCases);
        if(tensilicaCases.size() > 0){//Kalpesh,
SFDC-3225, Apr 04, 2016
```

```
TensilicaIPUtils.assignTensilicaCases(tensilicaCases);
    }
```

```
        // JIRA Issue SFDC-1408 || Prateek || END
        if(csCheckIPGMap.size() > 0){

            Map<Id, Boolean> csLicenseMap =
getValidEntitlementCheckForDIPCases(csCheckIPGMap.values());
            for(Id csId : csLicenseMap.keySet())
```

```

{
    if(!
    (csCheckIPGMap.get(csId).Contact_Employee_Flag__c &&
    csCheckIPGMap.get(csId).Parent_Account__c
    ==Label.cadence_design_system_id) && csLicenseMap.get(csId) ==
    false ){// Rashi ASR :640278 Dec 15,2014

    casesForIPG.add(csCheckIPGMap.get(csId));
    }
    }
    if(casesForIPG.size(>0){
        for(Case cs : casesForIPG){

            if(!
            (GlobalHelper.getRecordTypeIdtoName('Case') != null &&
            GlobalHelper.getRecordTypeIdtoName('Case').containsKey(cs.RecordType
            Id) &&
            GlobalHelper.getRecordTypeIdtoName('Case').get(cs.RecordTypeId) ==
            'DIP' && cs.Override_Reason__c != null)) {
                cs.ownerId =
                IPGQueueId;

            cs.Assignment_Status__c = ASSIGNED;

            cs.Assignment_Reason__c += 'IPG Support Assignment';
            }
            else
            if(GlobalHelper.getRecordTypeIdtoName('Case') != null &&
            GlobalHelper.getRecordTypeIdtoName('Case').containsKey(cs.RecordType
            Id) &&
            GlobalHelper.getRecordTypeIdtoName('Case').get(cs.RecordTypeId) ==
            'DIP' ){

            dipWithOverrideReasonMap.put(cs.id , true);
            }

        }

        System.debug('casesForIPG ::'+casesForIPG +
        'IPGQueueId :: '+IPGQueueId);
        //ASR 568432, Kalpesh Vyas, Jan 09, 2013 --
        END

        Set<String> productSet= new Set<String>();
        Set<String> productFeatureSet = new
        Set<String>();

        Set<String> accountSet = new Set<String>();
        Set<String> supportCenterSet = new

        Set<String> buSummarySet = new

        Set<String>();
        Set<Boolean> setItarCase = new Set<Boolean>();

```

```

// Project Mainstream ::: Abhishek ::: to
store the VAR Account IDs
Set<String> soldByAccountIdSet = new
Set<String>();

//Grab Product Feature Queues
for(Case caseRec: caseToAssign) {

caseProductQueues.put(caseRec.Product_Feature_Queue__c,null);
productFeatureList.add(caseRec.Product_Feature__c);
if(caseRec.Product_Feature_Lvl_1__c != null){
productFeatureListLevel1.add(caseRec.Product_Feature_Lvl_1__c);
}

if(caseRec.Product_Feature_Lvl_2__c != null){
productFeatureListLevel2.add(caseRec.Product_Feature_Lvl_2__c);
}

productFeatureList.add(caseRec.Product_Feature__c);
if(!
productClassList.contains(caseRec.Product__c)){
productClassList.add(caseRec.Product__c); //Kalpesh May 7, 2013, ASR
565916
}
// T-96472 – check Advanced Support
Group Id also

if(caseRec.Advanced_Support_Group_ID__c != null){
ASGIdSet.add(caseRec.Advanced_Support_Group_ID__c);
}

if(caseRec.Product__c != null)
productSet.add(caseRec.Product__c);
if(caseRec.Product_Feature__c !=
null)
productFeatureSet.add(caseRec.Product_Feature__c);
if(caseRec.Parent_Account__c !=
null)
accountSet.add(caseRec.Parent_Account__c);
if(caseRec.Support_Center__c != null
&& caseRec.Support_Center__c != '')

```





```

IP, :NO_CASE_OWNER
Ticket: 03299280
IN :productSupportQueues
IN :caseProductQueues.keySet()]);

System.debug ('groupList: ' + groupList);
Map<String, Id> groupOrUserNameMap = new
Map<String, Id>(); //Kalpesh May 1, 2013, ASR 565916

for(Group g : groupList.values()){
    groupOrUserNameMap.put(g.Name,
g.Id); //Kalpesh May 1, 2013, ASR 565916

    if(g.Name == SSG_QUEUE) SSGQueueId
= g.Id;
    if(g.Name == DELAY_QUEUE) delayQueue
= g.Id;
    if(g.Name == GCC_ICD_STD)
GCC_ICD_STDQueue = g.Id;
    if(g.Name == CS_TEMP) CS_TEMPQueue =
g.Id;
    if(g.Name == DIP) DIPQueue = g.Id;
    if(g.Name == NO_CASE_OWNER)
NO_CASE_OWNERQueue = g.Id;
    if(g.Name ==
GCC_ALLEGRO)GCC_AllegroQueue = g.Id; //Ticket: 03279500 Kalpesh,
Mar 25, 2013
    for (Advanced_Support_Groups__c asg:
asgList){
        if(g.Name == asg.Queue__c){
            System.debug('QUEUE
NAME' + asg.Queue__c);
productContactOverrideQueueMap.put(g.Name, g.Id);
        }
    }
    if
(caseProductQueues.containsKey(g.Name))
caseProductQueues.put(g.Name, g.Id);
}

system.debug('productSupportQueues ::'+productSupportQueues);

for(User supportUser :[SELECT Id, LastName,
Name FROM User
Where LastName = :CVA_SUPPORT OR Name = :VIP]){
    if(supportUser.LastName ==

```

```

CVA_SUPPORT){
                                CVASupportUserId =
supportUser.Id;
                                }else if(supportUser.Name == VIP){
                                VIPSupportUserId =
supportUser.Id;
                                }

groupOrUserNameMap.put(supportUser.Name, supportUser.Id);
                                }

system.debug('CVASupportUserId ::'+CVASupportUserId);

system.debug('groupOrUserNameMap ::'+groupOrUserNameMap);
                                //Do setup information

setupCacheInformation(ASGProdFeatMap,isASGJctQueryRun );

system.debug('productClassList ::'+productClassList);
                                //Kalpesh May 1, 2013, ASR 565916 -start--
                                Map<Id, QueueAssignment__c>
prodFeatQueueAssignmentMap = new Map<Id, QueueAssignment__c>();
                                Map<Id, QueueAssignment__c>
productQueueAssignmentMap = new Map<Id, QueueAssignment__c>();
                                Map<String, Map<String, Map<String,
Map<String, List<QueueAssignment__c>>>>>
supportCenterQueueAssignmentMap
                                = new Map<String, Map<String,
Map<String, Map<String, List<QueueAssignment__c>>>>>(); //Kalpesh,
Oct 29, 2015, SFDC-2089

System.debug('Product set-----'+productSet);
                                List<QueueAssignment__c> queueAssignmentList
= [SELECT Check_User_Availability__c, No_User_Found__c, Id,

                                Product__c,
Product_Feature__c, Name, Assignment_Reason__c,

                                Support_Center__c,
Account__c, BU_Summary__c, VAR_Account__c,ITAR_Case__c /* Project
Mainstream ::: Abhishek ::: */

                                FROM

QueueAssignment__c

                                WHERE

RecordType.Name='Case'

                                AND (Product__c

IN :productSet OR Product__c = null)

```

```

AND
(Product_Feature__c IN :productFeatureSet OR Product_Feature__c =
null)

AND (BU_Summary__c
IN :buSummarySet OR BU_Summary__c = null)

AND
(Support_Center__c IN :supportCenterSet OR Support_Center__c = null)

AND (VAR_Account__c
IN :soldByAccountIdSet OR VAR_Account__c = null) /* Project
Mainstream ::: Abhishek ::: added */
AND (ITAR_Case__c IN:setItarCase)

AND (Account__c
IN :accountSet OR Account__c = null)];

```

```

//Get Record Types for Case
Schema.DescribeSObjectResult caseSchema =
Schema.SObjectType.Case;
Map<String,Schema.RecordTypeInfo>
caseRecordTypeInfo = caseSchema.getRecordTypeInfoByNames();

System.debug('CASE List: ' + caseToAssign);
Integer caseIndx =
0; //Anjali : #03260706 #03260620
for(Case caseRec: caseToAssign) {

```

```

System.debug('Cases.productMap :::'+Cases.productMap);

System.debug('caseRec.Product_Class_Name__c
'+caseRec.Product_Class_Name__c
+',caseRec.Product__c '+
caseRec.Product__c);

System.debug('Cases.ProdToCVASupport.contains:::
'+Cases.ProdToCVASupport.contains(caseRec.Product_Class_Name__c));
//ASR 567346, Kalpesh July 17, 2013
START --
Boolean bypassJapanCase = false;

```

```

system.debug('caseRec.Support_Center__c
'+caseRec.Support_Center__c );

// Project Mainstream :::
Abhishek ::: added condition - "&&
caseRec.Product_Sold_By_Account__c == null" to perform queue
assignment for VAR Supported Cases
if(caseRec.Support_Center__c ==

```

```

SUPPORT_JAPAN && caseRec.Product_Sold_By_Account__c == null){
    bypassJapanCase = true;
}
//ASR 567346, Kalpesh July 17, 2013

END --

//Get Product name
if (Cases.productMap != null
    &&
Cases.productMap.containsKey(caseRec.Product__c)) { //Kalpesh Feb 16,
#432
    productClassNameStr =
Cases.productMap.get(caseRec.Product__c);
}
else
if(Cases.ProdToCVASupport.contains(caseRec.Product_Class_Name__c))
{ //Kalpesh Feb 16, #432
    productClassNameStr =
caseRec.Product_Class_Name__c;
}
else{
    productClassNameStr =
caseRec.Product_Class_Name__c;
}

//Kalpesh May 7, 2013, ASR 565916 --
start
Boolean checkUserAvailability =
true; // default should always go to normal assignment.
String noUserQueueId = null;
QueueAssignment__c queueAssignment =
null;
System.debug('queue assignment check-----'+queueAssignmentList);
if(queueAssignmentList != null &&
queueAssignmentList.size() > 0) {
    queueAssignment =
getMatchedQueueAssignment(queueAssignmentList , caseRec);
    if(queueAssignment != null){
checkUserAvailability = queueAssignment.Check_User_Availability__c;
String
noUserFoundQueue = queueAssignment.No_User_Found__c;
itarcaseassign=noUserFoundQueue;

if(noUserFoundQueue != null ) {

if(groupOrUserNameMap.containsKey(noUserFoundQueue)){

noUserQueueId = groupOrUserNameMap.get(noUserFoundQueue);
}
}

}
}
}

```

```

system.debug('checkUserAvailability :: '+checkUserAvailability +'
noUserQueueId :: '+noUserQueueId);
// DIP KDT Enterprise Projects
Assignment
System.debug('====KDT
Assignment===='+caseRec);

System.debug('====contactCopaIDMap===='+contactCopaIDMap+'====copaIdI
pConfigMap===='+copaIdIpConfigMap+'====dipProjectQueueMap===='+dipProj
ectQueueMap);

try{
    if(caseRec.RecordTypeId !=
null &&
Cases.recTypeMap.get(caseRec.RecordTypeId).trim().equalsIgnoreCase('
DIP'))
        &&
contactCopaIDMap.get(caseRec.ContactId) !=null &&
copaIdIpConfigMap.get(contactCopaIDMap.get(caseRec.ContactId)) !=
null
        &&
dipProjectQueueMap.get(copaIdIpConfigMap.get(contactCopaIDMap.get(ca
seRec.ContactId)).Source__c) !=null
        &&
caseRec.product_Feature2__c ==
copaIdIpConfigMap.get(contactCopaIDMap.get(caseRec.ContactId)).Produ
ct_Feature__c){
            caseRec.ownerId =
dipProjectQueueMap.get(copaIdIpConfigMap.get(contactCopaIDMap.get(ca
seRec.ContactId)).Source__c);

caseRec.Assignment_Status__c = ASSIGNED;

caseRec.Assignment_Reason__c += 'DIP
'+copaIdIpConfigMap.get(contactCopaIDMap.get(caseRec.ContactId)).Sou
rce__c+' Assignment';
        }
    }catch(exception excp){
        system.debug('**** EXCEPTION
****'+excp);

Exception_Handler.catchException(excp,NULL,NULL,'assignMultipleCases
','Case');//Exception handling
    }
//Kalpesh May 7, 2013, ASR 565916 --
end
//ASR 567346, Kalpesh July 17 --
bypass if Case support center is Japan
//
system.debug('caseRec.Advanced_Support_Flag1__c ::'+caseRec.Contact.
Advanced_Support_Group__r.Active__c);
//Check for Advanced Support Group
if (!bypassJapanCase &&
caseRec.Assignment_Status__c ==
CaseAssignmentEngine_AC.CHECK_ASSIGNMENT

```

```

caseRec.Advanced_Support_Flag1__c) {
    &&
    //If All products
    if
    (productContactOverrideQueueMap.get(caseRec.Advanced_Support_Queue__c
    ) == null) {

caseRec.Assignment_Reason__c += ' No ASG Queue Found ';
    }
    else {
        if
        (caseRec.Advanced_Support_All_Products__c) {

caseRec.OwnerId =
productContactOverrideQueueMap.get(caseRec.Advanced_Support_Queue__c)
;

caseRec.Assignment_Reason__c += ' All Products Assignment ';
caseRec.Assignment_Status__c = CaseAssignmentEngine_AC.ASSIGNED;
//T-86608 :
Anjali : set case type to E2E
// Move in
Cases
//
caseRec.Case_Type__c = 'E2E';
//Add to
send chatter
if
(caseRec.Advanced_Support_ChatterGroup__c != null){
asgCaseChatter.add(caseRec);

asgCaseChatterMap.put(caseIdx, caseRec); //Anjali :
#03260706 #03260620
}
}
else {
    if
    (foundProductASG(caseRec)) {

caseRec.OwnerId =
productContactOverrideQueueMap.get(caseRec.Advanced_Support_Queue__c)
;

caseRec.Assignment_Reason__c += 'ASG Product Assignment ';
caseRec.Assignment_Status__c = CaseAssignmentEngine_AC.ASSIGNED;
//
T-86608 : Anjali : set case type to E2E
//

```

```

Move in Cases
caseRec.Case_Type__c = 'E2E';
Add to send chatter
(caseRec.Advanced_Support_ChatterGroup__c != null) {
    asgCaseChatter.add(caseRec);
    asgCaseChatterMap.put(caseIndx, caseRec);
    #03260706 #03260620
}
}
}
}
//Kalpesh Apr 30, 2013 Change logic
//Put Advance support group
assignment prior to Product Assignment.
System.debug('caseRec.Product_Feature_Name__c ::'+caseRec.Product_Fe
ature_Name__c);
system.debug('bypassJapanCase
'+bypassJapanCase +'caseRec.Assignment_Status__c
'+caseRec.Assignment_Status__c+
'CVASupportUserId
'+CVASupportUserId+ 'productClassNameStr '+productClassNameStr
+'contains in cva ? '
+Cases.ProdToCVASupport.contains(productClassNameStr));
//ASR 567346, Kalpesh July 17 --
byPass if Case support center is Japan
//Kalpesh Feb 16, #432 (Assign to
CVA Support)
if (!bypassJapanCase &&
caseRec.Assignment_Status__c ==
CaseAssignmentEngine_AC.CHECK_ASSIGNMENT
&& CVASupportUserId != null
&& Cases.ProdToCVASupport.contains(productClassNameStr)){
    if(caseRec.Support_Center__c
== SUPPORT_KOREA){//ASR 567573, Kalpesh Jan 28, 2013
        if(!
dipWithOverrideReasonMap.containsKey(caseRec.id)) {
            caseRec.Assignment_Reason__c += 'No Case Owner Assignment';
            caseRec.Assignment_Status__c = NO_CASE_OWNER_STATUS;
            caseRec.OwnerId = NO_CASE_OWNERQueue;
        } else{

```



```

caseRec.ownerId = IPGQueueId;

caseRec.Assignment_Status__c = ASSIGNED;

caseRec.Assignment_Reason__c += 'IPG Support Assignment';
                                }
                                }else{

caseRec.Assignment_Reason__c += 'CVA Support Assignment';

caseRec.Assignment_Status__c = CaseAssignmentEngine_AC.ASSIGNED;
                                caseRec.OwnerId =
CVASupportUserId;
                                }
                                }

                                /****#SFDC-4707 Lookup Table Support
Level Assignment changes Start***
                                try{
                                if(LUTSupLevelMap.size())>0
&&
Cases.recTypeMap.get(caseRec.RecordTypeId).trim().equalsIgnoreCase('
Default') && caseRec.Case_Type__c != 'E2E')

LUTSupportLevelAssign(caseRec, LUTSupLevelMap, new List <Case>(),
(Map<Id , Product_Class__c >) Cases.caseProductMap,
Cases.contactMap);
                                }catch(exception e)
{System.debug('====Error Found ===='+e);}
                                /****#SFDC-4707 Lookup Table Support
Level Assignment changes End***

                                //ASR 567346, Kalpesh July 17 --
byPass if Case support center is Japan
                                //Check for Delay
                                if(!bypassJapanCase &&
caseRec.Assignment_Status__c ==
CaseAssignmentEngine_AC.CHECK_ASSIGNMENT
                                &&
configRec.Delay_Active__c == true
                                &&
(caseRec.Assignment_Made__c == false ||
(caseRec.Status=='New' &&
Cases.recTypeMap.get(caseRec.RecordTypeId).trim().equalsIgnoreCase('
Default') &&
((caseRec.OwnerId != null && !
String.valueOf(caseRec.OwnerId).startsWith('00G') &&
CaseAutoAssignment__c.getInstance(caseRec.OwnerId)!=null &&
CaseAutoAssignment__c.getInstance(caseRec.OwnerId).AutoAssignOnInser
t__c) || Cases.UsersHavingSSG_PS.contains(caseRec.OwnerId))))
                                &&
noUserQueueId == null) {//Kalpesh May 7, 2013, ASR 565916

```

```

if(caseRec.isLicenseDelayAvoid__c){//ASR 567842, Kalpesh Nov 07,
2013

caseRec.Assignment_Reason__c += 'Bypassing CS_DELAY --> for
Licences :'+byPassLicenceProductStatus;
                                }else{
                                caseRec =
checkForDelay(caseRec);
                                }
                                }

System.debug('caseRec.Assignment_Status__c
'+caseRec.Assignment_Status__c);
                                //ASR 567346, Kalpesh July 17 --
ignore check user availability if support center is Japan
                                //Normal Weighted Assignment
                                if ((bypassJapanCase ||
checkUserAvailability) && //Kalpesh May 7, 2013, ASR 565916

(caseRec.Assignment_Status__c ==
CaseAssignmentEngine_AC.CHECK_ASSIGNMENT
                                ||
caseRec.Assignment_Status__c ==
CaseAssignmentEngine_AC.DELAY_REASSIGN
                                ||
caseRec.Assignment_Status__c ==
CaseAssignmentEngine_AC.NO_CASE_OWNER_REASSIGN_STATUS
                                ||
caseRec.Assignment_Status__c ==
CaseAssignmentEngine_AC.FIND_ASSIGNMENT
                                || (caseRec.OwnerId ==
GCC_AllegroQueue && gccAllegroStatus.contains(caseRec.Status)))
{ //Ticket: 03279500 Kalpesh
                                System.debug('before Normal
Assignment ');
                                caseRec =
doNormalAssignment(caseRec);
                                }

system.debug('caseRec.OwnerId ::'+caseRec.OwnerId);

system.debug('isNormalAssignmentFailed ::'+isNormalAssignmentFailed)
;

                                // ASR 639628 Devaki Oct 17 2014 start
                                if(caseRec.OwnerId != null &&
String.valueOf(caseRec.OwnerId).startsWith('00G')) {
                                caseRec.Manager__c = null;
                                }
                                // ASR 639628 Devaki Oct 17 2014
end
                                //ASR 567346, Kalpesh July 17 --
byPass if Case support center is Japan

```

```

//Kalpesh May 7, 2013, ASR 565916 --
start
    if(!bypassJapanCase &&
noUserQueueId != null &&
        (checkUserAvailability ?
(isNormalAssignmentFailed != null ? isNormalAssignmentFailed :
false)
        :
(caseRec.Assignment_Status__c ==
CaseAssignmentEngine_AC.CHECK_ASSIGNMENT)) && !
dipWithOverrideReasonMap.containsKey(caseRec.id)){
        system.debug('inside product
support assignment');
        caseRec.OwnerId =
noUserQueueId;
        caseRec.Assignment_Status__c
= CaseAssignmentEngine_AC.ASSIGNED;
        caseRec.Assignment_Reason__c
+= queueAssignment.Assignment_Reason__c;
    }
//Kalpesh May 7, 2013, ASR 565916 --
end

//Set the Assigned Date if assigned.
if (caseRec.Assignment_Status__c ==
CaseAssignmentEngine_AC.ASSIGNED
    ||
caseRec.Assignment_Status__c !=
CaseAssignmentEngine_AC.NO_CASE_OWNER_STATUS
    ||
caseRec.Assignment_Status__c != CaseAssignmentEngine_AC.IN_DELAY
    ||
caseRec.Assignment_Status__c !=
CaseAssignmentEngine_AC.IN_DELAY_CS_TEMP){
        caseRec.Assigned_Date__c =
DateTime.Now();
        caseRec.Assignment_Made__c =
true;
    }
    caseIndx +
+; //Anjali : #03260706 #03260620
}

/*if(ownerSet != null && ownerSet.size() >
0){
        ownerMap =
Cases.fetchOwnerManager(ownerSet, false);
    }*/
    if(caseToUpdateManager != null &&
caseToUpdateManager.size() > 0){
        for(Case caseRec:
caseToUpdateManager) {

```

```
if(ownerSet.contains(caseRec.OwnerID)){  
    && ownerMap.containsKey(caseRec.OwnerID)){  
        = ownerMap.get(caseRec.OwnerID);  
  
caseRec.Manager__c = null;  
  
    }  
}  
  
}
```

```
if(ownerMap != null  
    caseRec.Manager__c  
}else {  
  
}
```

```

/**
 * @description Find out the queue for the case
 * @param queueAssignmentList List of Queue assignment object
 * @param cs CAsE object for which the queue needs to be
checked
 * @return QueueAssignment__c The matched queueAssignment
object is returned
 */

```

```

        public static QueueAssignment__c
getMatchedQueueAssignment( List<QueueAssignment__c>
queueAssignmentList , Case cs){
    Integer maxMatchCount = -1;
    Integer maxMatchScore = -1;
    QueueAssignment__c matchQueueAssignment = null;
    for(QueueAssignment__c qa : queueAssignmentList ){
        Integer matchCount = 0;
        Integer matchScore = 0;
        System.debug('before the condition for itar'+cs.ITARCase__c +
qa.ITAR_Case__c);
        if(qa.ITAR_Case__c && cs.ITARCase__c ==
qa.ITAR_Case__c ){
            matchCount++;
            matchScore += 5;
        } else if(qa.ITAR_Case__c && cs.ITARCase__c !=
qa.ITAR_Case__c){
            matchCount = -1;
            break;
        }
        if(!qa.ITAR_Case__c && cs.ITARCase__c ==
qa.ITAR_Case__c ){
            matchCount++;
            matchScore += 5;
        } else if(!qa.ITAR_Case__c && cs.ITARCase__c !=
qa.ITAR_Case__c){
            matchCount = -1;
            break;
        }
    }
}

```

```

        if(qa.Product__c != null && cs.Product__c ==
qa.Product__c){
            matchCount++;
            matchScore += 5;
        } else if(qa.Product__c != null &&
cs.Product__c != qa.Product__c){
            matchCount = -1;
            break;
        }

        if(qa.Product_Feature__c != null &&
cs.Product_Feature__c == qa.Product_Feature__c ){
            matchCount++;
            matchScore += 4;
        } else if(qa.Product_Feature__c != null &&
cs.Product_Feature__c != qa.Product_Feature__c ){
            matchCount = -1;
            break;
        }

        if(qa.Account__c != null && cs.Parent_Account__c
== qa.Account__c){
            matchCount++;
            matchScore += 3;
        } else if(qa.Account__c != null &&
cs.Parent_Account__c != qa.Account__c){
            matchCount = -1;
            break;
        }

```

```

        // Project Mainstream ::: Abhishek ::: --
START
        if(qa.VAR_Account__c != null &&
cs.Product_Sold_By_Account__c == qa.VAR_Account__c){
            matchCount++;
            matchScore += 3;
        } else if(qa.VAR_Account__c != null &&
cs.Product_Sold_By_Account__c != qa.VAR_Account__c){
            matchCount = -1;
            break;
        }
        // Project Mainstream ::: Abhishek ::: --
END

```

```

        if(qa.Support_Center__c != null &&
cs.Support_Center__c == qa.Support_Center__c ){
            matchCount++;
            matchScore += 2;
        } else if(qa.Support_Center__c != null &&
cs.Support_Center__c != qa.Support_Center__c ){
            matchCount = -1;
            break;
        }

```

```

        if(qa.BU_Summary__c != null &&
cs.BU_Summary_Attribute__c == qa.BU_Summary__c ){
            matchCount++;
            matchScore += 1;
        } else if(qa.BU_Summary__c != null &&
cs.BU_Summary_Attribute__c != qa.BU_Summary__c ){
            matchCount = -1;
            break;
        }

        if(maxMatchCount < matchCount
            || (maxMatchCount == matchCount &&
maxMatchScore < matchScore)) {
            matchQueueAssignment = qa;
            maxMatchCount = matchCount;
            maxMatchScore = matchScore;
        }
    }
    return matchQueueAssignment;
}

/**
 * @description Method to setup SSG Code
 * @param caseList List of case objects
 */
public static void createSSGQueuedTask(List<Case> caseList){

    /*      Map<String,Schema.RecordTypeInfo> rtQueuedMapByName
            =
Schema.SObjectType.Queued_Task__c.getRecordTypeInfosByName();

        for(Case caseRec: caseList) {

            Queued_Task__c queuedTaskRec = new
Queued_Task__c();

            if (caseRec.Origin == 'COS') {
                queuedTaskRec.Activity_Type__c =
'COS Case';
            }
            else {
                if (caseRec.Subject == 'Service
Request Submission') {
                    queuedTaskRec.Activity_Type__c = 'COS Case Failure';
                }
                if (caseRec.Subject == 'Attn: SSG/
CRC - COS account created') {
                    queuedTaskRec.Activity_Type__c = 'COS Creation';
                }
            }
        }
    }

```

```

// T-105782 - Already added this condition
If Case.Product = "Design IP"
    if (caseRec.Product_Class_Name__c ==
CaseStaticData.DIP_Product && caseRec.ContactId == NULL) {
        queuedTaskRec.Activity_Type__c =
'DIP Contact';
    }
    // T-105782 - If Case.Contact = NULL
    else if(caseRec.ContactId == NULL &&
(caseRec.Product_Class_Name__c == 'Memory Model'
||
caseRec.Product_Class_Name__c == SIGRITY_PRODUCT)){
        queuedTaskRec.Activity_Type__c =
'Contact Creation Request';
    }
    // ASR-567845 : Jitendar;Email Inbound
Queued Task Should not be created for DIP cases
    if (queuedTaskRec.Activity_Type__c == NULL){
        if(caseRec.Product_Class_Name__c !=
CaseStaticData.DIP_Product) {

queuedTaskRec.Activity_Type__c = 'Email Inbound';
        }else {
            continue;
        }
    }
    //queuedTaskRec.Account__c =
caseRec.AccountId;
    queuedTaskRec.Case__c = caseRec.Id;
    queuedTaskRec.Contact__c =
caseRec.ContactId;
    queuedTaskRec.Name = 'Case #' +
caseRec.CaseNumber + ' ' + queuedTaskRec.Activity_Type__c;
    queuedTaskRec.Product_Class__c =
caseRec.Product__c;
    queuedTaskRec.RecordTypeId =
rtQueuedMapByName.get('SSG').getRecordTypeId();
    queuedTaskRec.Case_To_Email_Address__c =
caseRec.To_Email_Address__c;

    queuedTaskList.add(queuedTaskRec);

}   SFDC-13490 Commented */

// SFDC-13490 Code Start
Map<String,Schema.RecordTypeInfo> rtQueuedMapByName
=
Schema.SObjectType.Task.getRecordTypeInfoByName();

for(Case caseRec: caseList) {

    Task queuedTaskRec = new Task();

```

```

        if (caseRec.Origin == 'COS') {
            queuedTaskRec.Activity_Type__c =
'COS Case';
        }
        else {
            if (caseRec.Subject == 'Service
Request Submission') {
                queuedTaskRec.Activity_Type__c = 'COS Case Failure';
            }
            if (caseRec.Subject == 'Attn: SSG/
CRC - COS account created') {
                queuedTaskRec.Activity_Type__c = 'COS Creation';
            }
        }

        // T-105782 - Already added this condition
If Case.Product = "Design IP"
        if (caseRec.Product_Class_Name__c ==
CaseStaticData.DIP_Product && caseRec.ContactId == NULL) {
            queuedTaskRec.Activity_Type__c =
'DIP Contact';
        }
        // T-105782 - If Case.Contact = NULL
        else if(caseRec.ContactId == NULL &&
(caseRec.Product_Class_Name__c == 'Memory Model'
||
caseRec.Product_Class_Name__c == SIGRITY_PRODUCT)){
            queuedTaskRec.Activity_Type__c =
'Contact Creation Request';
        }
        // ASR-567845 : Jitendar;Email - Inbound
Queued Task Should not be created for DIP cases
        if (queuedTaskRec.Activity_Type__c == NULL){
            if(caseRec.Product_Class_Name__c !=
CaseStaticData.DIP_Product) {

                queuedTaskRec.Activity_Type__c = 'Email - Inbound';
            }else {
                continue;
            }
        }
        //queuedTaskRec.Account__c =
caseRec.AccountId;

        queuedTaskRec.Case__c = caseRec.Id;
        queuedTaskRec.WhoId = caseRec.ContactId;
        queuedTaskRec.Subject = 'Case #' +
caseRec.CaseNumber + ' ' + queuedTaskRec.Activity_Type__c;
        queuedTaskRec.Product_Class__c =
caseRec.Product__c;

        queuedTaskRec.RecordTypeId =

```



```

rtQueuedMapByName.get('SSG').getRecordTypeId();
        queuedTaskRec.Case_To_Email_Address__c =
caseRec.To_Email_Address__c;

        queuedTaskList.add(queuedTaskRec);

    }
    // SFDC-13490 Code End

    System.debug('queuedTaskList:::' + queuedTaskList);
    //Insert Queued Task
    if (queuedTaskList.size() > 0 ) upsert
queuedTaskList;
    }

    /
    ****
    ***
        Method to create ASG Chatter posts on Insert

    ****
    ***/
        public static void logChatterToGroupPostInsert(List<Case>
externalCaseList){

            //Find Groups
            List<String> chatterGroupNameList = new
List<String>();
            for (Case caseTemp: externalCaseList){

chatterGroupNameList.add(caseTemp.Advanced_Support_ChatterGroup__c);
            }

            System.debug('chatterGroupNameList' +
chatterGroupNameList);

            if (chatterGroupNameList.size() > 0 ) {
                List<CollaborationGroup> chatterGroup =
                    [SELECT Id, Description,
CollaborationType, Name FROM CollaborationGroup
                    WHERE Name
IN :chatterGroupNameList];

                List<FeedItem> feedItemList = new
List<FeedItem>();
                Integer caseIndx = 0; //Anjali : #03260706
#03260620
                for (Case caseTemp: externalCaseList){

                    if (caseTemp.CaseNumber!= NULL &&
asgCaseChatterMap != null &&
asgCaseChatterMap.containsKey(caseIndx)) { //Anjali : #03260706
#03260620

                        FeedItem post = new

```

```

FeedItem();

        post.LinkUrl =
System.Label.Case_Base_URL + '/' + caseTemp.Id;
        post.Body = 'Severity: ' +
caseTemp.Priority;
        post.Body = post.Body +
'\nProduct: ' + caseTemp.Product_Class_Name__c;
        post.Body = post.Body +
'\nProduct Feature: ' + caseTemp.Product_Feature_Name__c;
        post.Body = post.Body +
'\nContact Name: ' + caseTemp.Contact_Name__c;
        post.Title = 'Case #' +
caseTemp.CaseNumber;

        for (CollaborationGroup
collabGroupRec: chatterGroup) {
            if
(collabGroupRec.Name == caseTemp.Advanced_Support_ChatterGroup__c)
post.ParentId = collabGroupRec.Id;
        }
        if ( post.ParentId != NULL)
feedItemList.add(post);
        }
        caseIndx++; //Anjali : #03260706
#03260620
    }

    if (feedItemList.size() > 0) insert
feedItemList;
    }

}

/**
 * @description Kalpesh, SFDC-4205, Dec 30, 2016, Separating
list based on PF, PF1 and PF2
 * @param productSupportAgentRec product Support agent
 * @param groupMemberPF2List group member and PF2 level
mapping
 * @param groupMemberPF1List group member and PF1 level
mapping
 * @param groupMemberPFLList group member and PF mapping
 */

private static void
setGroupMemberlist(Product_Support_Agent__c productSupportAgentRec,
Map<Id, Product_Support_Agent__c> groupMemberPF2List,

        Map<Id, Product_Support_Agent__c>
groupMemberPF1List, Map<Id, Product_Support_Agent__c>
groupMemberPFLList){

```

```

if(productSupportAgentRec.Product_feature_level_2__c != null){

groupMemberPF2List.put(productSupportAgentRec.User__c,
productSupportAgentRec);
    }else
if(productSupportAgentRec.Product_Feature_level_1__c != null){

groupMemberPF1List.put(productSupportAgentRec.User__c,
productSupportAgentRec);
    }else{

groupMemberPFLList.put(productSupportAgentRec.User__c,
productSupportAgentRec);
    }
}

/**
 * @description Method to setup SSG Code
 * @param ASGProdFeatMap Collection of ASG and Product Feature
 * @param isASGJctQueryRun
 */
public static void setupCacheInformation(Map<String,
Set<String>> ASGProdFeatMap,boolean isASGJctQueryRun ){

    //Setup Normal Assignment
    scoreMap = new map<Id, integer>();
    ownerMap = new map<String, String>();
    //Kalpesh Vyas, Feb 22, 2016, SFDC-2820
    prodFeatLevel1UserExpertiseMap = new Map<Id, Map<Id,
String>>();
    prodFeatLevel2UserExpertiseMap = new Map<Id, Map<Id,
String>>();
    prodFeatLevel1UserSupportCenters = new Map<Id,
Map<Id, Set<String>>>(); //pflvl1 -> user -> support centers
    prodFeatLevel2UserSupportCenters = new Map<Id,
Map<Id, Set<String>>>(); //pflvl2 -> user -> support centers

    System.debug('productFeatureListLevel2
'+productFeatureListLevel2 + ',productFeatureListLevel1 '+
productFeatureListLevel1
                                +',productFeatureList
'+productFeatureList);
    system.debug('case rec ----'+itarcaseassign);

    supportAgentMap = new map<Id,
Product_Support_Agent__c>
                                ([select
user__c,user__r.ManagerId, expertise__c, Product_Feature__c,
Support_Center__c,
                                User__r.Region__c,
Product_Feature_level_1__c, Product_Feature_level_2__c, Name
                                from

```

```

Product_Support_Agent__c
                                where
((Product_Feature_level_2__c IN :productFeatureListLevel2 AND
Product_Feature_level_2__c !=null) OR

(Product_Feature_level_1__c IN :productFeatureListLevel1 AND
Product_Feature_level_1__c !=null AND Product_Feature_level_2__c =
null) OR

                                (Product_Feature__c
IN :productFeatureList AND Product_Feature__c != NULL AND
Product_Feature_level_2__c = null AND Product_Feature_level_1__c =
null))//SFDC-2820, Manish Shuk

                                AND Available__c =
true AND User__r.Available_For_Support__c = true
                                AND User__r.IsActive
= true AND User__r.Employment_Status__c = 'Available'];

System.debug('SupportAgentMappp---'+supportAgentMap);

                                for (Product_Support_Agent__c
productSupportAgentRec : supportAgentMap.values()) {
                                scoreMap.put(productSupportAgentRec.User__c,
0);
                                if(productSupportAgentRec.User__c != null &&
productSupportAgentRec.User__r.ManagerId != null){

ownerMap.put(productSupportAgentRec.User__c,
productSupportAgentRec.User__r.ManagerId);
                                }

                                if(productSupportAgentRec.User__c != null &&
productSupportAgentRec.User__r.Region__c != null
                                &&
productSupportAgentRec.User__r.Region__c == GCS){//Ticket: 03279500
Kalpesh //Jyoti Goyal, JIRA issue SFDC-1595,28/4/15
                                //
groupNormalGCCAllegroMemeberList.put(productSupportAgentRec.User__c,
productSupportAgentRec);

setGroupMemberlist(productSupportAgentRec,
groupNormalPFL2GCCAllegroMemeberList,
groupNormalPFL1GCCAllegroMemeberList,
groupNormalGCCAllegroMemeberList);
                                //*** #SFDC-4709 GCS AE Assignment
changes ***

if(productSupportAgentRec.Support_Center__c ==
SUPPORT_NORTH_AMERICA)setGroupMemberlist(productSupportAgentRec,
groupNormalPFL2GCSNAMemberList, groupNormalPFL1GCSNAMemberList,
groupNormalGCSNAMemberList);
                                }
                                //***#SFDC-4709 NA Critical Advance Support

```

Cases Changes Start\*\*\*

```
        if(productSupportAgentRec.User__c != null &&
productSupportAgentRec.User__r.Region__c != null
        &&
productSupportAgentRec.User__r.Region__c == NACS &&
productSupportAgentRec.Support_Center__c == SUPPORT_NORTH_AMERICA){
```

```
    setGroupMemberlist(productSupportAgentRec,
groupNormalPFL2NACSMemberList, groupNormalPFL1NACSMemberList,
groupNormalNACSMemberList);
```

```
    }//If ends
```

```
    /***#SFDC-4709 NA Critical Advance Support
```

Cases Changes End\*\*\*

```
        if
(productSupportAgentRec.Support_Center__c == SUPPORT_CHINA){
```

```
    setGroupMemberlist(productSupportAgentRec,
groupNormalPFL2ChinaMemberList, groupNormalPFL1ChinaMemberList,
groupNormalChinaMemberList);
```

```
    }
```

```
        if
```

```
(productSupportAgentRec.Support_Center__c == SUPPORT_FRANCE){
```

```
    setGroupMemberlist(productSupportAgentRec,
groupNormalPFL2FranceMemberList, groupNormalPFL1FranceMemberList,
groupNormalFranceMemberList);
```

```
    }
```

```
        if
```

```
(productSupportAgentRec.Support_Center__c == SUPPORT_GERMANY){
```

```
    setGroupMemberlist(productSupportAgentRec,
groupNormalPFL2GermanyMemberList, groupNormalPFL1GermanyMemberList,
groupNormalGermanyMemberList);
```

```
    }
```

```
        if
```

```
(productSupportAgentRec.Support_Center__c == SUPPORT_HONG_KONG){
```

```
    setGroupMemberlist(productSupportAgentRec,
groupNormalPFL2HongKongMemberList,
groupNormalPFL1HongKongMemberList, groupNormalHongKongMemberList);
```

```
    }
```

```
        if
```

```
(productSupportAgentRec.Support_Center__c == SUPPORT_INDIA){
```

```
    setGroupMemberlist(productSupportAgentRec,
groupNormalPFL2IndiaMemberList, groupNormalPFL1IndiaMemberList,
groupNormalIndiaMemberList);
```

```
    }
```

```
        if
```

```
(productSupportAgentRec.Support_Center__c == SUPPORT_ISREAL){
```

```
    setGroupMemberlist(productSupportAgentRec,
groupNormalPFL2IsrealMemberList, groupNormalPFL1IsrealMemberList,
```

```

groupNormalIsrealMemberList);
    }
    if
(productSupportAgentRec.Support_Center__c == SUPPORT_JAPAN){

setGroupMemberlist(productSupportAgentRec,
groupNormalPFL2JapanMemberList, groupNormalPFL1JapanMemberList,
groupNormalJapanMemberList);
    }
    if
(productSupportAgentRec.Support_Center__c == SUPPORT_KOREA){

setGroupMemberlist(productSupportAgentRec,
groupNormalPFL2KoreaMemberList, groupNormalPFL1KoreaMemberList,
groupNormalKoreaMemberList);
    }
    if
(productSupportAgentRec.Support_Center__c == SUPPORT_NORTH_AMERICA){

setGroupMemberlist(productSupportAgentRec,
groupNormalPFL2NorthAmericaMemberList,
groupNormalPFL1NorthAmericaMemberList,
groupNormalNorthAmericaMemberList);
    }
    if
(productSupportAgentRec.Support_Center__c == SUPPORT_RUSSIA){

setGroupMemberlist(productSupportAgentRec,
groupNormalPFL2RussiaMemberList, groupNormalPFL1RussiaMemberList,
groupNormalRussiaMemberList);
    }
    if
(productSupportAgentRec.Support_Center__c == SUPPORT_SINGAPORE){

setGroupMemberlist(productSupportAgentRec,
groupNormalPFL2SingaporeMemberList,
groupNormalPFL1SingaporeMemberList, groupNormalSingaporeMemberList);
    }
    if
(productSupportAgentRec.Support_Center__c == SUPPORT_TAIWAN){

setGroupMemberlist(productSupportAgentRec,
groupNormalPFL2TaiwanMemberList, groupNormalPFL1TaiwanMemberList,
groupNormalTaiwanMemberList);
    }
    if
(productSupportAgentRec.Support_Center__c == SUPPORT_UNITED_KINGDOM)
{

setGroupMemberlist(productSupportAgentRec,
groupNormalPFL2UnitedKingdomMemberList,
groupNormalPFL1UnitedKingdomMemberList,
groupNormalUnitedKingdomMemberList);
    }
}

```

```

        if
(productSupportAgentRec.Support_Center__c == SUPPORT_UNITED_STATES){

setGroupMemberList(productSupportAgentRec,
groupNormalPFL2UnitedStatesMemberList,
groupNormalPFL1UnitedStatesMemberList,
groupNormalUnitedStatesMemberList);
        }

    }

    //supportAgentExpertiseMap<agent<productfeature,
expertise>>
supportAgentExpertiseMap = new map<Id, map<id,
String>>());

    for(Product_Support_Agent__c productSupportRec:
supportAgentMap.values()){

        //Kalpesh Vyas, Feb 22, 2016, SFDC-2820, --
START
        //Assuming each Pf level 2 will always have
pf level 1 and pf level 1 have pf

        //Adding logic for pf lvl1 and pf lvl 2 for
//Kalpesh Vyas, Dec 15, 2016, SFDC-4205
changed sequence and conditions.

if(productSupportRec.Product_Feature_level_2__c != null &&
productFeatureListLevel2.contains(productSupportRec.Product_Feature_
level_2__c)){

        if(!
prodFeatLevel2UserExpertiseMap.containsKey(productSupportRec.Product
_Feature_level_2__c)){

prodFeatLevel2UserExpertiseMap.put(productSupportRec.Product_Feature
_level_2__c, new Map<Id, String>());

prodFeatLevel2UserSupportCenters.put(productSupportRec.Product_Featu
re_level_2__c, new Map<Id, Set<String>>());
        }

prodFeatLevel2UserExpertiseMap.get(productSupportRec.Product_Feature
_level_2__c).put(productSupportRec.User__c,
productSupportRec.Expertise__c);

        if(!
prodFeatLevel2UserSupportCenters.get(productSupportRec.Product_Featu
re_level_2__c).containsKey(productSupportRec.User__c)){

prodFeatLevel2UserSupportCenters.get(productSupportRec.Product_Featu
re_level_2__c).put(productSupportRec.User__c, new Set<String>());
        }

```

```

prodFeatLevel2UserSupportCenters.get(productSupportRec.Product_Feature_level_2__c).get(productSupportRec.User__c).add(productSupportRec.Support_center__c);
    }else
    if(productSupportRec.Product_Feature_level_2__c == null &&
productSupportRec.Product_Feature_level_1__c != null &&
productFeatureListLevel1.contains(productSupportRec.Product_Feature_level_1__c)){
        if(!
prodFeatLevel1UserExpertiseMap.containsKey(productSupportRec.Product_Feature_level_1__c)){

prodFeatLevel1UserExpertiseMap.put(productSupportRec.Product_Feature_level_1__c,

                                new Map<Id, String>());

prodFeatLevel1UserSupportCenters.put(productSupportRec.Product_Feature_level_1__c, new Map<Id, Set<String>>());
        }

prodFeatLevel1UserExpertiseMap.get(productSupportRec.Product_Feature_level_1__c).put(productSupportRec.User__c,
productSupportRec.Expertise__c);
        if(!
prodFeatLevel1UserSupportCenters.get(productSupportRec.Product_Feature_level_1__c).containsKey(productSupportRec.User__c)){

prodFeatLevel1UserSupportCenters.get(productSupportRec.Product_Feature_level_1__c).put(productSupportRec.User__c, new Set<String>());
        }

prodFeatLevel1UserSupportCenters.get(productSupportRec.Product_Feature_level_1__c).get(productSupportRec.User__c).add(productSupportRec.Support_center__c);
    }else
    if(productSupportRec.Product_Feature_level_2__c == null &&
productSupportRec.Product_Feature_level_1__c == null){
        if(!
supportAgentExpertiseMap.containsKey(productSupportRec.User__c)){

supportAgentExpertiseMap.put(productSupportRec.User__c,
                                new map<id,
String>{productSupportRec.Product_Feature__c =>
productSupportRec.Expertise__c});

        }else{

supportAgentExpertiseMap.get(productSupportRec.User__c).put(productSupportRec.Product_Feature__c, productSupportRec.Expertise__c);

        }
    }
}
//Kalpesh Vyas, Feb 22, 2016, SFDC-2820, --

```



END

}

```
List<Case> regularSupportCaseList = [select Id,
OwnerId, Priority, Status, Product_Feature__c
                                from Case where
isClosed = false
                                AND Status IN
('New', 'Transfer', 'Wait/CCR', 'Wait/Critical CCR',
                                'Wait/Cust', 'Wait/
Expert', 'Wait/Release', 'Working')
                                AND OwnerId
IN :scoreMap.keySet()
                                order by
CDN_Last_Modified_Date__c DESC LIMIT 1000]; //replacing
LastModifiedDate with CDN_Last_Modified_Date__c: ASR 569057

for(Case regularSupportCaseRec:
regularSupportCaseList){
    integer caseMultiplier = 0;
    caseMultiplier =
getCaseMultiplier(regularSupportCaseRec);

scoreMap.put(regularSupportCaseRec.OwnerId, (integer)
scoreMap.get(regularSupportCaseRec.OwnerId)
            + 1 * caseMultiplier);
//+
getExpertiseNumber(regularSupportCaseRec.Priority,
supportAgentExpertiseMap.get(regularSupportCaseRec.ownerId).get(regu
larSupportCaseRec.Product_Feature__c));
}

//Setup Advanced Support Group
// T-96472 - optimize the Advanced Support Group set
up
if(ASGProdFeatMap == null || !isASGJctQueryRun ){
    Set<String> productFeatureIds = new
Set<String>();

productFeatureIds.addAll((List<String>)productFeatureList);
    asgProdFeatIdsMap =
CaseService.fetchASGProductFeatures(productFeatureIds, ASGIdSet);
}else{
    asgProdFeatIdsMap = ASGProdFeatMap;
}
/*asgProdFeatList = [SELECT Product_Feature__c,
Advanced_Support_Group__c FROM
                                ASG_to_Prod_Feat_Junc__c WHERE
Product_Feature__c IN :productFeatureList];

*/
}
```

```

        private static Integer getCaseMultiplier(Case
regularSupportCaseRec){
            Integer caseMultiplier = 0;
            if (regularSupportCaseRec.Status == 'New')
caseMultiplier = (integer) configRec.New__c;
            if (regularSupportCaseRec.Status == 'Transfer')
caseMultiplier = (integer) configRec.Transfer__c;
            if (regularSupportCaseRec.Status == 'Wait/CCR')
caseMultiplier = (integer) configRec.Wait_CCR__c;
            if (regularSupportCaseRec.Status == 'Wait/Critical
CCR') caseMultiplier = (integer) configRec.Wait_Critical_CCR__c;
            if (regularSupportCaseRec.Status == 'Wait/Cust')
caseMultiplier = (integer) configRec.Wait_Cust__c;
            if (regularSupportCaseRec.Status == 'Wait/Expert')
caseMultiplier = (integer) configRec.Wait_Expert__c;
            if (regularSupportCaseRec.Status == 'Wait/Release')
caseMultiplier = (integer) configRec.Wait_Release__c;
            if (regularSupportCaseRec.Status == 'Working')
caseMultiplier = (integer) configRec.Working__c;
            return caseMultiplier;
        }

    /**
    @description Get Expertise Number
    @param priorityStr Priority of the case
    @param expertiseStr Expertise of the AE
    @return Integer Returns the expertise number based on the
combination of priority and expertise
    */
    private static integer getExpertiseNumber(String
priorityStr, String expertiseStr) {

        if (priorityStr == 'Critical' && expertiseStr ==
'Expert')
            return
(integer)configRec.Critical_Expert__c;
        if (priorityStr == 'Critical' && expertiseStr ==
'Intermediate')
            return
(integer)configRec.Critical_Intermediate__c;
        if (priorityStr == 'Critical' && expertiseStr ==
'Novice')
            return
(integer)configRec.Critical_Novice__c;
        if (priorityStr == 'Important' && expertiseStr ==
'Expert')
            return
(integer)configRec.Important_Expert__c;
        if (priorityStr == 'Important' && expertiseStr ==
'Intermediate')
            return
(integer)configRec.Important_Intermediate__c;
        if (priorityStr == 'Important' && expertiseStr ==
'Novice')

```

```

        return
(integer)configRec.Important_Novice__c;
        if (priorityStr == 'Minor' && expertiseStr ==
'Expert')
                return
(integer)configRec.Minor_Expert__c;
        if (priorityStr == 'Minor' && expertiseStr ==
'Intermediate')
                return
(integer)configRec.Minor_Intermediate__c;
        if (priorityStr == 'Minor' && expertiseStr ==
'Novice')
                return
(integer)configRec.Minor_Novice__c;
        return 0;
}

/
*****
***
        Get Expertise String

*****
***/
        /*private static string getExpertiseString(Id
productFeature, Id agentId) {
                /*String expertiseStr = '';
                for(Product_Support_Agent__c productSupportRec:
supportAgentMap.values()){
                        if (productSupportRec.User__c == agentId
&&
productSupportRec.Product_Feature__c == productFeature)
                                expertiseStr =
productSupportRec.Expertise__c;
                }*/
                /*return
supportAgentExpertiseMap.get(agentId).get(productFeature);
                //return expertiseStr;
        }*/

        /**
        @description Advance Support Group Check
        @param caseRec Case in which ASG is supposed to be checked
        @return boolean Returns whether the case contains ASG or not
        */
        private static boolean foundProductASG(Case caseRec){
                system.debug('asgProdFeatIdsMap
'+asgProdFeatIdsMap );

system.debug( 'CaseRec.Advanced_Support_Group_ID__c ::'+CaseRec.Advan
ced_Support_Group_ID__c

```

```

+'CaseRec.Product_Feature__c ::'+CaseRec.Product_Feature__c);

system.debug('asgProdFeatIdsMap.containsKey(CaseRec.Advanced_Support_Group_ID__c)'+asgProdFeatIdsMap.containsKey(CaseRec.Advanced_Support_Group_ID__c));
// T-96472 - optimize the code for found Product
ASG

if(asgProdFeatIdsMap.containsKey(CaseRec.Advanced_Support_Group_ID__c) &&
asgProdFeatIdsMap.get(CaseRec.Advanced_Support_Group_ID__c).contains(CaseRec.Product_Feature__c)){
    return true;
}
return false;
//T-110700 : use 15 digit key
/* if(CaseRec.Advanced_Support_Group_ID__c != null){
    String asgId =
String.valueOf(CaseRec.Advanced_Support_Group_ID__c).substring(0,
15);
    if(asgProdFeatIdsMap.containsKey(asgId) &&
asgProdFeatIdsMap.get(asgId).contains(CaseRec.Product_Feature__c)){
        return true;
    }
}
return false;*/

/*for (ASG_to_Prod_Feat_Junc__c asgJunctRec:
asgProdFeatList)
    if (asgJunctRec.Advanced_Support_Group__c ==
CaseRec.Advanced_Support_Group_ID__c
        && asgJunctRec.Product_Feature__c ==
CaseRec.Product_Feature__c ) {
        return true;
    }
return false;*/

}

/
*****
***
    @description Normal weighted assignment logic
    @param caseRec Case record for which the assignment is
supposed to be done
    @return Case Returns the case object with updated values
related to case assignment

*****
***/

private static Case doNormalAssignment(Case caseRec){

    System.debug('Normal Assignment Case:: '+caseRec);
System.debug('Normal Assignment Case with potential queue::

```

```
'+itarcaseassign);
```

```
        caseRec.Assignment_Reason__c += ' NORMAL PROCESS ';
        Boolean isGCCAllegroQueue = caseRec.OwnerId ==
GCC_AllegroQueue;
        Boolean isNACS
=(Cases.recTypeMap.get(caseRec.RecordTypeId).trim().equalsIgnoreCase
('Default') && CaseRec.Support_Level__c == 'Advanced support' &&
caseRec.Support_Center__c == SUPPORT_NORTH_AMERICA &&
caseRec.Priority == 'Critical' ? true : false);//#SFDC-4709 changes
        Boolean isGCS = (!isNACS &&
Cases.recTypeMap.get(caseRec.RecordTypeId).trim().equalsIgnoreCase('
Default') && (caseRec.Assignment_Status__c ==
CaseAssignmentEngine_AC.IN_DELAY || caseRec.Assignment_Status__c ==
CaseAssignmentEngine_AC.DELAY_REASSIGN) ? true : false);//#SFDC-4709
changes
        System.debug('==Checking for isGCS : '+ isGCS +'==
Checking for isNACS : '+ isNACS);

        if (caseRec.Assignment_Status__c ==
CaseAssignmentEngine_AC.DELAY_REASSIGN && !isNACS) { // #SFDC-4709
changes
            caseRec.Assignment_Status__c =
CaseAssignmentEngine_AC.IN_DELAY_CS_TEMP;
            //ASR 567602, Kalpesh August 16, 2013 START
            try{
                Case_Assignment_Interval__c csDelay
= Case_Assignment_Interval__c.getAll().get('CS_TEMP_DELAY');
                if(csDelay.Hours__c == null &&
csDelay.Minutes__c == null){

caseRec.Delay_Assignment_Date__c =
DateTime.Now().addHours(CS_TEMP_DELAY_TIME);
                }else{
                    Decimal hours=0, minutes=0;
                    if(csDelay.Hours__c != null)
hours = csDelay.Hours__c;
                    if(csDelay.Minutes__c !=
null) minutes = csDelay.Minutes__c;

caseRec.Delay_Assignment_Date__c =
DateTime.Now().addHours(hours.intValue()).addMinutes(minutes.intValu
e());
                }
            }catch(Exception ex){
                caseRec.Delay_Assignment_Date__c =
DateTime.Now().addHours(CS_TEMP_DELAY_TIME);
            }
            //ASR 567602, Kalpesh August 16, 2013 END
            caseRec.OwnerId = CS_TEMPQueue;
        }
        else if(!isGCCAllegroQueue && !isGCS) {//Ticket: 03279500
```

```

Kalpesh, Mar 25, 2013 //SFDC-4709 changes
        if(!
dipWithOverrideReasonMap.containsKey(caseRec.id)) {
            caseRec.Assignment_Status__c =
CaseAssignmentEngine_AC.NO_CASE_OWNER_STATUS;
            caseRec.OwnerId =
NO_CASE_OWNERQueue;
        } else{
            caseRec.ownerId = IPGQueueId;
            caseRec.Assignment_Status__c =
ASSIGNED;
            caseRec.Assignment_Reason__c += 'IPG
Support Assignment';
        }
    }

    System.debug('caseProductQueues: ' +
caseProductQueues);
    if(caseRec.Product_Feature_Queue__c != ''
        &&
caseProductQueues.containsKey(caseRec.Product_Feature_Queue__c)
        &&
caseProductQueues.get(caseRec.Product_Feature_Queue__c) != NULL) {
        caseRec.OwnerId =
caseProductQueues.get(caseRec.Product_Feature_Queue__c);
        caseRec.Assignment_Reason__c += ' Product
Feature Queue Assigned ';
        caseRec.Assignment_Status__c =
CaseAssignmentEngine_AC.ASSIGNED;
    }
    else {
        Map<Id, Product_Support_Agent__c>
normalCurrentMemberList = new Map<Id, Product_Support_Agent__c>();
        Map<Id, Product_Support_Agent__c>
normalPFL1CurrentMemberList = new Map<Id,
Product_Support_Agent__c>();
        Map<Id, Product_Support_Agent__c>
normalPFL2CurrentMemberList = new Map<Id,
Product_Support_Agent__c>();
        //Default to Contact and look for incoming
mail support center
        String supportCenter =
caseRec.Support_Center__c;

        if(isGCCAllegroQueue){//Ticket: 03279500
Kalpesh, Mar 25, 2013
            normalCurrentMemberList =
groupNormalGCCAllegroMemeberList;
            normalPFL1CurrentMemberList =
groupNormalPFL1GCCAllegroMemeberList;
            normalPFL2CurrentMemberList =
groupNormalPFL2GCCAllegroMemeberList;
        }
        /* else if(!isGCCAllegroQueue) {//Ticket: 03279500

```

Kalpesh, Mar 25, 2013

```
        caseRec.Assignment_Status__c =
CaseAssignmentEngine_AC.NO_CASE_OWNER_STATUS;
        caseRec.OwnerId = NO_CASE_OWNERQueue;
    } */
    else{

        if (supportCenter == NULL ) {
            if (caseRec.Origin == 'Email
- China') supportCenter= SUPPORT_CHINA;
            if (caseRec.Origin == 'Email
- France') supportCenter= SUPPORT_FRANCE;
            if (caseRec.Origin == 'Email
- Germany') supportCenter= SUPPORT_GERMANY;
            //if (caseRec.Origin ==
'Email - Israel') supportCenter= SUPPORT_ISREAL;
            if (caseRec.Origin == 'Email
- Japan') supportCenter= SUPPORT_JAPAN;
            if (caseRec.Origin == 'Email
- Korea') supportCenter= SUPPORT_KOREA;
            //if (caseRec.Origin ==
'Email - Russia') supportCenter= SUPPORT_RUSSIA;
            if (caseRec.Origin == 'Email
- Singapore') supportCenter= SUPPORT_SINGAPORE;
            if (caseRec.Origin == 'Email
- Taiwan') supportCenter= SUPPORT_TAIWAN;
            if (caseRec.Origin == 'Email
- India') supportCenter= SUPPORT_INDIA;

        }

        if (supportCenter == NULL ) {
            supportCenter =
SUPPORT_NORTH_AMERICA;

        }

        //Default if Department of
Defense //ASR 639345, Kalpesh, July 24, 2014 START
        if(caseRec.Parent_Account__c !=
null ){

            System.debug('caseRec.Parent_Account__c
'+caseRec.Parent_Account__c);

            CaseAssignmentRelatedValues__c
valueOfDOD = CaseAssignmentRelatedValues__c.getInstance(DOD);
            if(valueOfDOD != null &&
valueOfDOD.ValueList__c!=null){

                if(caseRec.Parent_Account__c == valueOfDOD.ValueList__c.trim())
                    supportCenter =
SUPPORT_UNITED_STATES;

            }else if(caseRec.Contact != null
&& caseRec.Contact.Parent_Account__r.Name != null
&&
```

```

caseRec.Contact.Parent_Account__r.Name == DOD){
    supportCenter =
SUPPORT_UNITED_STATES;
    }
}
//ASR 639345, Kalpesh, July 24, 2014

```

END

```

    if (supportCenter ==
SUPPORT_NORTH_AMERICA) {
        normalCurrentMemberList =
groupNormalNorthAmericaMemberList;
        normalPFL1CurrentMemberList
= groupNormalPFL1NorthAmericaMemberList;
        normalPFL2CurrentMemberList
= groupNormalPFL2NorthAmericaMemberList;
    }
    if (supportCenter == SUPPORT_CHINA)
{
        normalCurrentMemberList =
groupNormalChinaMemberList;
        normalPFL1CurrentMemberList
= groupNormalPFL1ChinaMemberList;
        normalPFL2CurrentMemberList
= groupNormalPFL2ChinaMemberList;
    }
    if (supportCenter ==
SUPPORT_FRANCE) {
        normalCurrentMemberList =
groupNormalFranceMemberList;
        normalPFL1CurrentMemberList
= groupNormalPFL1FranceMemberList;
        normalPFL2CurrentMemberList
= groupNormalPFL2FranceMemberList;
    }
    if (supportCenter ==
SUPPORT_GERMANY) {
        normalCurrentMemberList =
groupNormalGermanyMemberList;
        normalPFL1CurrentMemberList
= groupNormalPFL1GermanyMemberList;
        normalPFL2CurrentMemberList
= groupNormalPFL2GermanyMemberList;
    }
    if (supportCenter == SUPPORT_INDIA)
{
        normalCurrentMemberList =
groupNormalIndiaMemberList;
        normalPFL1CurrentMemberList
= groupNormalPFL1IndiaMemberList;
        normalPFL2CurrentMemberList
= groupNormalPFL2IndiaMemberList;
    }
    if (supportCenter == SUPPORT_ISREAL)

```



```

{
    groupNormalIsrealMemberList;
    = groupNormalPFL1IsrealMemberList;
    = groupNormalPFL2IsrealMemberList;
    }
    if (supportCenter == SUPPORT_JAPAN)
    {
        groupNormalJapanMemberList;
        = groupNormalPFL1JapanMemberList;
        = groupNormalPFL2JapanMemberList;
    }
    if (supportCenter == SUPPORT_KOREA)
    {
        groupNormalKoreaMemberList;
        = groupNormalPFL1KoreaMemberList;
        = groupNormalPFL2KoreaMemberList;
    }
    if (supportCenter == SUPPORT_RUSSIA)
    {
        groupNormalRussiaMemberList;
        = groupNormalPFL1RussiaMemberList;
        = groupNormalPFL2RussiaMemberList;
    }
    if (supportCenter ==
SUPPORT_SINGAPORE) {
        groupNormalSingaporeMemberList;
        = groupNormalPFL1SingaporeMemberList;
        = groupNormalPFL2SingaporeMemberList;
    }
    if (supportCenter == SUPPORT_TAIWAN)
    {
        groupNormalTaiwanMemberList;
        = groupNormalPFL1TaiwanMemberList;
        = groupNormalPFL2TaiwanMemberList;
    }
    if (supportCenter ==

```

```

SUPPORT_UNITED_KINGDOM) {
    normalCurrentMemberList =
groupNormalUnitedKingdomMemberList;
    normalPFL1CurrentMemberList
= groupNormalPFL1UnitedKingdomMemberList;
    normalPFL2CurrentMemberList
= groupNormalPFL2UnitedKingdomMemberList;
    }
    if (supportCenter ==
SUPPORT_UNITED_STATES) {
        normalCurrentMemberList =
groupNormalUnitedStatesMemberList;
        normalPFL1CurrentMemberList
= groupNormalPFL1UnitedStatesMemberList;
        normalPFL2CurrentMemberList
= groupNormalPFL2UnitedStatesMemberList;
    }

    /****#SFDC-4709 GCS AE Assignment
and NA Critical Advance Support Cases Changes Start***
    if(isNACS){
        normalCurrentMemberList =
groupNormalNACSMemberList;
        normalPFL1CurrentMemberList
= groupNormalPFL1NACSMemberList;
        normalPFL2CurrentMemberList
= groupNormalPFL2NACSMemberList;
        System.debug('====I AM IN
NACS ASSIGNMENT CHECK====');//Log
    }//If ends
    if(isGCS){
        normalCurrentMemberList =
groupNormalGCSNAMemberList;
        normalPFL1CurrentMemberList
= groupNormalPFL1GCSNAMemberList;
        normalPFL2CurrentMemberList
= groupNormalPFL2GCSNAMemberList;
        System.debug('====I AM IN
GCS ASSIGNMENT CHECK====');//Log
    }//If ends
    /****#SFDC-4709 GCS AE Assignment
and NA Critical Advance Support Cases Changes End***

    System.debug('supportCenter
'+supportCenter + 'case support center '+
caseRec.Support_Center__c);

    System.debug('supportAgentExpertiseMap '+supportAgentExpertiseMap);

    System.debug('normalCurrentMemberList '+normalCurrentMemberList );

    System.debug('normalPFL1CurrentMemberList
'+normalPFL1CurrentMemberList );

```

```

System.debug('normalPFL2CurrentMemberList
'+normalPFL2CurrentMemberList );

        }
        id ownerIdAssign = null;
        integer lowestScore = null;
        map<Id, integer> debugScoreMap = new map<Id,
integer>();

        System.debug('scoreMap ::'+scoreMap);
        if(caseRec.Id != null) System.debug('case
Id'+caseRec.Id );

                integer addedCaseExpertNumber = 0;
itarApprovedUserIds= getItarApprovedUserIds(supportAgentMap);
System.debug('isItarApprovedUser-----'+itarApprovedUserIds);

                //check map of prod feat level2 for case
prodfeatlevel2 and check their available support agents
                if(caseRec.Product_Feature_Lvl_2__c != null
&& prodFeatLevel2UserExpertiseMap.size() > 0 &&
prodFeatLevel2UserExpertiseMap.containsKey(caseRec.Product_Feature_L
vl_2__c)){
                        lowestScore = null;
                        debugScoreMap = new map<Id,
integer>();
                        System.debug('In Level 2
assignment');
                                for(Id agentId :
prodFeatLevel2UserExpertiseMap.get(caseRec.Product_Feature_Lvl_2__c)
.keySet()){
                                        //Kalpesh, Mar 29, 2016, if
does not match with Case support center ignore.
                                        if(!
prodFeatLevel2UserSupportCenters.get(caseRec.Product_Feature_Lvl_2__
c).get(agentId).contains(caseRec.Support_center__c))
                                                continue;

                                                addedCaseExpertNumber =
getExpertiseNumber(caseRec.Priority,
prodFeatLevel2UserExpertiseMap.get(caseRec.Product_Feature_Lvl_2__c)
.get(agentId));
                                                System.debug('agentId ::
'+agentId +', addedCaseExpertNumber:: '+addedCaseExpertNumber);

System.debug('normalPFL2CurrentMemberList.containsKey(agentId) '+
normalPFL2CurrentMemberList.containsKey(agentId));

                                if
(normalPFL2CurrentMemberList.containsKey(agentId) &&
normalPFL2CurrentMemberList.get(agentId).Product_Feature_level_2__c

```

```

== caseRec.Product_Feature_Lvl_2__c && caseRec.ITARCase__c== false)
{ //Kalpesh, SFDC-4205, Dec 30, 2016

debugScoreMap.put(agentId, scoreMap.get(agentId)+
addedCaseExpertNumber);

                                                                    if
(lowestScore == null) {

                                                                    lowestScore = scoreMap.get(agentId) +
addedCaseExpertNumber;
                                                                    }
                                                                    if
(scoreMap.get(agentId) + addedCaseExpertNumber <= lowestScore) {

                                                                    ownerIdAssign = agentId;

                                                                    lowestScore = scoreMap.get(agentId) + addedCaseExpertNumber;
                                                                    }

System.debug('ownerIdAssign '+ownerIdAssign +', lowestScore
'+lowestScore);
}

                                                                    else if
(normalPFL2CurrentMemberList.containsKey(agentId) &&
normalPFL2CurrentMemberList.get(agentId).Product_Feature_level_2__c
== caseRec.Product_Feature_Lvl_2__c && caseRec.ITARCase__c== true &&
itarApprovedUserIds.containsKey(agentId)) {

debugScoreMap.put(agentId, scoreMap.get(agentId)+
addedCaseExpertNumber);

                                                                    if
(lowestScore == null) {

                                                                    lowestScore = scoreMap.get(agentId) +
addedCaseExpertNumber;
                                                                    }
                                                                    if
(scoreMap.get(agentId) + addedCaseExpertNumber <= lowestScore) {

                                                                    ownerIdAssign = agentId;

                                                                    lowestScore = scoreMap.get(agentId) + addedCaseExpertNumber;
                                                                    }

System.debug('ownerIdAssign '+ownerIdAssign +', lowestScore
'+lowestScore);

                                                                    }
                                                                    }if(ownerIdAssign == null &&
CaseRec.Product_Feature_Lvl_1__c != null &&
prodFeatLevel1UserExpertiseMap.size() > 0 &&

prodFeatLevel1UserExpertiseMap.containsKey(caseRec.Product_Feature_L
vl_1__c)){

```

```

System.debug('In Level 1 assignment');
                                lowestScore = null;
                                debugScoreMap = new map<Id,
integer>();
                                for(Id agentId :
prodFeatLevel1UserExpertiseMap.get(caseRec.Product_Feature_Lvl_1__c)
.keySet()){
                                //Kalpesh, Mar 29, 2016, if
does not match with Case support center ignore.
                                if(!
prodFeatLevel1UserSupportCenters.get(caseRec.Product_Feature_Lvl_1__
c).get(agentId).contains(caseRec.Support_center__c))
                                continue;

                                addedCaseExpertNumber =
getExpertiseNumber(caseRec.Priority,
prodFeatLevel1UserExpertiseMap.get(caseRec.Product_Feature_Lvl_1__c)
.get(agentId));
                                System.debug('agentId ::
'+agentId +', addedCaseExpertNumber:: '+addedCaseExpertNumber);

                                //
caseRec.Assignment_Reason__c += ' agentId: ' + agentId + '=' +
addedCaseExpertNumber;

                                if
(normalPFL1CurrentMemberList.containsKey(agentId)
                                &&
normalPFL1CurrentMemberList.get(agentId).Product_Feature_level_1__c
== caseRec.Product_Feature_Lvl_1__c && caseRec.ITARCase__c== false )
{ //Kalpesh, SFDC-4205, Dec 30, 2016

debugScoreMap.put(agentId, scoreMap.get(agentId)+
addedCaseExpertNumber);

                                if (lowestScore ==
null) {
                                lowestScore =
scoreMap.get(agentId) + addedCaseExpertNumber;
                                }
                                if
(scoreMap.get(agentId) + addedCaseExpertNumber <= lowestScore){
ownerIdAssign = agentId;

lowestScore = scoreMap.get(agentId) + addedCaseExpertNumber;
                                }

System.debug('ownerIdAssign '+ownerIdAssign +'lowestScore
'+lowestScore);
                                }
else if (normalPFL1CurrentMemberList.containsKey(agentId)

```

```

normalPFL1CurrentMemberList.get(agentId).Product_Feature_level_1__c
== caseRec.Product_Feature_Lvl_1__c && caseRec.ITARCase__c== true &&
itarApprovedUserIds.contains(agentId) ) { //Kalpesh, SFDC-4205, Dec
30, 2016

```

```

debugScoreMap.put(agentId, scoreMap.get(agentId)+
addedCaseExpertNumber);

```

```

null) {
    if (lowestScore ==
        lowestScore =
scoreMap.get(agentId) + addedCaseExpertNumber;
    }
    if
(scoreMap.get(agentId) + addedCaseExpertNumber <= lowestScore){
ownerIdAssign = agentId;
lowestScore = scoreMap.get(agentId) + addedCaseExpertNumber;
    }

```

```

System.debug('ownerIdAssign '+ownerIdAssign +'lowestScore
'+lowestScore);
    }

```

```

}

```

```

    }if(ownerIdAssign == null &&
supportAgentExpertiseMap.size()>0){//SFDC-4205, Kalpesh Dec 13, 2016
        lowestScore = null;
        debugScoreMap = new map<Id,
integer>();
        for (Id agentId: scoreMap.keySet())
        {
            System.debug('In PF
assignment');
            //Kalpesh, SFDC-2820, Feb
22, 2016
            //Adding logic to check for
pf lvl 2 availablity and then check for pf lvl 1 avaiablity and
then earlier pf logic
            System.debug('agentId
'+agentId);
            //SFDC-4205, Kalpesh Dec 13,
2016
            if(!
supportAgentExpertiseMap.containsKey(agentId) || !
supportAgentExpertiseMap.get(agentId).containsKey(caseRec.Product_Fe
ature__c)) continue;//Kalpesh, Mar 24, 2016, SFDC-2820

```

```

        addedCaseExpertNumber =

```

```

getExpertiseNumber(caseRec.Priority,
supportAgentExpertiseMap.get(agentId).get(caseRec.Product_Feature__c
));
                                System.debug('agentId ::
'+agentId +', addedCaseExpertNumber:: '+addedCaseExpertNumber);
                                //
caseRec.Assignment_Reason__c += ' agentId: ' + agentId + '=' +
addedCaseExpertNumber;
                                if
(normalCurrentMemberList.containsKey(agentId) &&
caseRec.ITARCase__c== false) { //Kalpesh, SFDC-4205, Dec 30, 2016

debugScoreMap.put(agentId, scoreMap.get(agentId)+
addedCaseExpertNumber);
                                if (lowestScore ==
null) {

                                lowestScore = scoreMap.get(agentId) + addedCaseExpertNumber;
                                }
                                if
(scoreMap.get(agentId) + addedCaseExpertNumber <= lowestScore) {

ownerIdAssign = agentId;

lowestScore = scoreMap.get(agentId) + addedCaseExpertNumber;
                                }

System.debug('ownerIdAssign '+ownerIdAssign +'lowestScore
'+lowestScore);
                                }
                                else if
(normalCurrentMemberList.containsKey(agentId) &&
caseRec.ITARCase__c== true && itarApprovedUserIds.contains(agentId))
{ //Kalpesh, SFDC-4205, Dec 30, 2016

debugScoreMap.put(agentId, scoreMap.get(agentId)+
addedCaseExpertNumber);
                                if (lowestScore ==
null) {

                                lowestScore = scoreMap.get(agentId) + addedCaseExpertNumber;
                                }
                                if
(scoreMap.get(agentId) + addedCaseExpertNumber <= lowestScore) {

ownerIdAssign = agentId;

lowestScore = scoreMap.get(agentId) + addedCaseExpertNumber;
                                }

System.debug('ownerIdAssign '+ownerIdAssign +'lowestScore
'+lowestScore);

```

```

    }

    }

    if(isGCCAllegroQueue){
        caseRec.Assignment_Reason__c += '
'+GCC_ALLEGRO_ASSIGNMENT + ' MAP: ' + debugScoreMap;
    }else{
        //caseRec.Assignment_Reason__c += '
' + supportCenter + ' MAP: ' + debugScoreMap;
        caseRec.Assignment_Reason__c += ' '
+ (isNACS ? NACS_ASSIGNMENT : (isGCS ? GCS_ASSIGNMENT :
supportCenter)) + ' MAP: ' + debugScoreMap; //SFDC-4709 changes
    }

    if (caseRec.ITARCase__c== true && ownerIdAssign != null) {
        if (caseRec.ITARCase__c== true && !
itarApprovedUserIds.contains(ownerIdAssign)) {
            caseRec.Assignment_Reason__c += ' Case cannot be
assigned because agent is not in ITAR queue.';
            isNormalAssignmentFailed = true;

            System.debug('inside thew new method for itarcase');
        } else {
            caseRec.OwnerId = ownerIdAssign;
            caseRec.Assignment_Status__c =
CaseAssignmentEngine_AC.ASSIGNED;
            Integer caseMultiplier = getCaseMultiplier(caseRec);
            scoreMap.put(ownerIdAssign, (integer)
scoreMap.get(ownerIdAssign) + 1 * caseMultiplier);
            isNormalAssignmentFailed = false;
            System.debug('inside the else new method for
itarcase');
        }
    }

    else if (ownerIdAssign != null) {
        caseRec.OwnerId = ownerIdAssign;
        caseRec.Assignment_Status__c =
CaseAssignmentEngine_AC.ASSIGNED;
        Integer caseMultiplier =
getCaseMultiplier(caseRec);
        // addedCaseExpertNumber is replaced
with caseMultiplier
        scoreMap.put(ownerIdAssign,
(integer) scoreMap.get(ownerIdAssign)
+ 1 * caseMultiplier);

        if(String.valueOf(caseRec.OwnerId).startsWith('005')){
            caseRec.Hidden_Owner__c =

```



```

caseRec.OwnerId;

ownerSet.add(caseRec.OwnerId);

caseToUpdateManager.add(caseRec);
    }

        else {
            caseRec.Hidden_Owner__c =
null;    // ASR 639628 Devaki Oct 17 2014
            caseRec.Manager__c =
null;    // ASR 639628 Devaki Oct 17 2014
        }
        isNormalAssignmentFailed = false;
    }
    else {
        //D0 SSG Assigned if nobody assigned
        caseRec.Assignment_Reason__c += '
NORMAL ASSIGNMENT FAILED.  ';
        //caseRec.Assignment_Status__c =
CaseAssignmentEngine_AC.NO_CASE_OWNER_STATUS;
        //caseRec =
doSSGAssignment(caseRec);
        //caseRec.OwnerId = CS_TEMPQueue;
        isNormalAssignmentFailed = true;
    }
}

return caseRec;
}

/**
    @description to get the itar approved user from this
method
    @return Boolean value Case Returns the case with updated
values related to case assignment
    */
    public static Set<Id> getItarApprovedUserIds(Map<Id,
Product_Support_Agent__c> supportAgentMap) {
        itarApprovedUserIds = new Set<Id>();

        // Query for matching queues based on the 'itarcaseassign'
value.
        List<Group> matchingQueues = [SELECT Id FROM Group WHERE
Type = 'Queue' AND Name = :itarcaseassign LIMIT 1];
        System.debug('queue Id-----' + matchingQueues);

        Set<Id> itarQueueIdSet = new Set<Id>();
        for (Group queue : matchingQueues) {
            itarQueueIdSet.add(queue.Id);
        }

        if (itarQueueIdSet != null && !itarQueueIdSet.isEmpty() &&
itarQueueIdSet.size() > 0) {
            // Retrieve ITAR queue members using the provided

```

method.

```
        Map<Id, List<Id>> itarQueueMemberMap =
EmailService.getQueueMemberIDs(itarQueueIdSet);
        System.debug('Value of queue-----' +
itarQueueMemberMap);

        for (Product_Support_Agent__c supportAgent :
supportAgentMap.values()) {
            if (!itarQueueMemberMap.isEmpty() &&
itarQueueMemberMap.values() != null) {
                for (List<Id> memberIDs :
itarQueueMemberMap.values()) {
                    if
(memberIDs.contains(supportAgent.User__c)) {
itarApprovedUserIds.add(supportAgent.User__c);
                    }
                }
            }
        }

        return itarApprovedUserIds;
    }
}
```

```
    /**
    @description Check for Delay for the case that is being
processed
    @param caseRec the case for whih system need to check for
delay
    @return Case Returns the case with updated values related
to case assignment
    */
    private static Case checkForDelay(Case caseRec){

        //SFDC-4708 Get the account type value from Custom
setting
        Set<String> accSet =
Utility_Account.getAccountType(NO_DELAY);

        caseRec.Assignment_Reason__c += 'Checked for Delay.
';

        //Do not delay if no Contact
        if (caseRec.ContactId == NULL) {
            caseRec.Assignment_Reason__c += 'NODELAY ->
No Contact';
            return caseRec;
        }
    }
}
```

```

//ASR 567878, Kalpesh Oct 23, 2013
//updated logic with ticket 03425177
//updated logic Ticket : 03440637, Kalpesh, Dec
18,2013
        if((caseRec.Support_Center__c ==
SUPPORT_UNITED_STATES || caseRec.Support_Center__c ==
SUPPORT_NORTH_AMERICA) &&
            (( caseRec.Priority == 'Critical' &&
CaseRec.Support_Level__c == 'Premium')
            ||( CaseRec.Support_Level__c ==
'Advanced Support' && (caseRec.Priority == 'Critical' ||
caseRec.Priority == 'Important')))){

                                caseRec.Assignment_Reason__c += '
NODELAY -> 1) Case.Seveity equals '+caseRec.Priority+', Case Support
Level equals '+ CaseRec.Support_Level__c+', Case Support Center
equals  '+caseRec.Support_Center__c;
                                /*** #SFDC-4709 Delay queue Changes
End***

                                return caseRec;
        }

        //2) BU Summary in ("Hosted Design Solutions" ,
"Design IP", "SSG Solutions" , "Cadence Shared Tools")
        if (caseRec.BU_Summary_Attribute__c == 'Hosted
Design Solutions' ||
                                caseRec.BU_Summary_Attribute__c == 'Design
IP' ||
                                caseRec.BU_Summary_Attribute__c == 'SSG
Solutions' ||
                                caseRec.BU_Summary_Attribute__c == 'Cadence
Shared Tools') {
                                caseRec.Assignment_Reason__c +=
'NODELAY -> 2) BU Summary in ("Hosted Design Solutions" , "Design
IP" , "SSG Solutions" , "Cadence Shared Tools";
                                return caseRec;
        }

        //3) Case.Site Support Level equals "internal"
        if (CaseRec.Support_Level__c == 'Internal') {
                                caseRec.Assignment_Reason__c +=
'NODELAY -> 3) Case.Site Support Level equals "internal";
                                return caseRec;
        }

        //4) Contact Support Center not equal to "North
America"
        if(caseRec.Support_Center__c != 'North America') {
                                caseRec.Assignment_Reason__c +=
'NODELAY -> 4) Contact Support Center not equal to "North America";
                                return caseRec;
        }

        //5) Case.Source equals "Cust Solution"

```

```

        if(caseRec.Origin == 'Cust Solution') {
            caseRec.Assignment_Reason__c +=
'NODELAY -> 5) Case.Source equals "Cust Solution" ';
            return caseRec;
        }

        //6) Case.Site Support Level equals "Advanced
Support" and Case.Severity equals "2-Important"
        caseRec.Assignment_Reason__c += ' Support Level: ' +
caseRec.Support_Level__c;

        caseRec.Assignment_Reason__c += ' caseRec.Priority '
+ caseRec.Priority;
        if (caseRec.Support_Level__c == 'Advanced Support'
            && ( caseRec.Priority == 'Important' ||
caseRec.Priority == 'Critical') ) {
            caseRec.Assignment_Reason__c +=
'NODELAY -> 6) Case.Site Support Level equals "Advanced Support"
and Case.Severity equals "2-Important"';
            return caseRec;
        }

        //7) Account Type in ( Non-Alliance Competitor,
Industry Alliance)
        //SFDC-4708 Update the account type criteria based
on custom setting rather than hard coded Account type values
        //SFDC-5350 removed caseRec.Contact.Account_Type__c
field and added Parent_Account_Type__c field
        caseRec.Assignment_Reason__c += '
caseRec.Contact.Parent_Account_Type__c: ' +
caseRec.Contact.Parent_Account_Type__c;
        if (accSet.size() > 0 &&
accSet.contains(caseRec.Contact.Parent_Account_Type__c))
        {
            String acctypes;
            for(String accty:accSet)
            {
                if(acctypes == null)
                {
                    acctypes = accty;
                }
                else
                {
                    acctypes += ',' +
accty;
                }
            }
            caseRec.Assignment_Reason__c += '
NODELAY -> 7) Account Type in (' + acctypes + ') ';

            return caseRec;
        }
    }

```

```
//Record Type of VIP should not delay
if
(Cases.recTypeMap.get(caseRec.RecordTypeId).trim().equalsIgnoreCase('VIP')) {
    caseRec.Assignment_Reason__c +=
'NODELAY -> 9) VIP Case does not delay. ';
    return caseRec;
}
//Kalpesh Feb 13, 2013 (#386)(Assign to CVA Support
palladium XP product is also in the list)
//Kalpesh Feb 16, #432 (Hardware Platform tools if
not assigned to CVA Support then also not allowed to delay)
if(Cases.ProdToCVASupport.contains(caseRec.Product_Class_Name__c)){
    caseRec.Assignment_Reason__c += 'NODELAY ->
10) Case with Hardware Platform tools does not delay.';
    return caseRec;
}

caseRec.Assignment_Status__c =
CaseAssignmentEngine_AC.IN_DELAY;
//***#SFDC-4709 GCS AE Assignment cases Changes
Start***
try{
    if
(Cases.recTypeMap.get(caseRec.RecordTypeId).trim().equalsIgnoreCase('Default')){
        System.debug('====I AM IN GCS
IMMEDIATE ASSIGNMENT===='); //Log
        caseRec =
doNormalAssignment(caseRec);
        if(caseRec.Assignment_Status__c ==
CaseAssignmentEngine_AC.ASSIGNED)return caseRec;
    } //If Ends
} catch(Exception e){System.debug('=====Error Found
in Delay===='+e);}
//***#SFDC-4709 GCS AE Assignment cases Changes
End***

if(delayQueue != null)
    caseRec.OwnerId = delayQueue;
caseRec.Assignment_Reason__c += 'DELAY OF CASE';
try{
    Case_Assignment_Interval__c csDelay
= Case_Assignment_Interval__c.getAll().get('CS DELAY');
    if(csDelay.Hours__c == null &&
csDelay.Minutes__c == null){
caseRec.Delay_Assignment_Date__c =
DateTime.Now().addHours(CS_DELAY_TIME);
    }else{
        Decimal hours=0, minutes=0;
        if(csDelay.Hours__c != null)
```

```

hours = csDelay.Hours__c;
if(csDelay.Minutes__c !=
null) minutes = csDelay.Minutes__c;

caseRec.Delay_Assignment_Date__c =
DateTime.Now().addHours(hours.intValue()).addMinutes(minutes.intValue());
    }
    }catch(Exception ex){
        caseRec.Delay_Assignment_Date__c =
DateTime.Now().addHours(CS_DELAY_TIME);
    }
    // caseRec.Delay_Assignment_Date__c =
DateTime.Now().addHours(18);

    return caseRec;

}

/**
 * @description #SFDC-4707 Assign Lookup Table Support Level
values for Cases from assignMultipleCases and onBeforeInsert method
only for Default Cases
 * @param caseRecData The case for whcih the assessment has
to be done
 * @param resLUTMap The collection of LUT tables fetched
 * @param caseListData the complete case list
 * @param prodPltformMap product and platform map
 * @param contactMap The collection of contacts
 */
Public static void LUTSupportLevelAssign(Case caseRecData,
Map<String,Lookup_Table__c> resLUTMap, List <Case> caseListData, Map
< Id, Product_Class__c > prodPltformMap, Map < string,Contact>
contactMap){
    // Variable declarations
    List < Case > caseList = new List < Case > ();

    try{

if(resLUTMap.size(>0))System.debug('====resLUTMap===='+resLUTMap);//
/Log
        // Assign the single case or list of case
records to caseList variable
        if(caseRecData!=null &&
caseRecData.ContactId!=null)caseList.add(caseRecData);

if(caseListData.size(>0))caseList.addAll(caseListData);

        if(resLUTMap.size(>0)){
            for (Case caseRec: caseList){
                System.debug('====Case
Data===='+caseRec);//Log

```

```

if(Cases.recTypeMap.get(caseRec.RecordTypeId).trim().equalsIgnoreCase('Default') && caseRec.Case_Type__c != 'E2E')//
caseRec.Assignment_Status__c ==
CaseAssignmentEngine_AC.CHECK_ASSIGNMENT
{
// Check the Lookup
table support level value exit for case
if(caseRec.id!=null
&& caseRec.Contact.Support_Center__c!=null &&
caseRec.Contact.Parent_Account__c!=null &&
caseRec.Product__r.Product_Attribute__r.Cadence_Platform__c!=null )
{

System.debug('====Case ID
Found===='+caseRec.Contact.Support_Center__c+caseRec.Contact.Parent_
Account__c+caseRec.Product__r.Product_Attribute__r.Cadence_Platform_
__c);//Log
// Assign
the Lookup table support level value to case Support_Level__c field
if(!
caseRec.Ecommerce_Case__c)

if(resLUTMap.get(caseRec.Contact.Support_Center__c+caseRec.Contact.P
arent_Account__c+caseRec.Product__r.Product_Attribute__r.Cadence_Pla
tform__c)!
=null)caseRec.Support_Level__c=resLUTMap.get(caseRec.Contact.Support
_Center__c+caseRec.Contact.Parent_Account__c+caseRec.Product__r.Prod
uct_Attribute__r.Cadence_Platform__c).Support_Level__c;
}//If ends
else{

if(contactMap.size(>0))System.debug('====Case ID Blank
contactMap===='+contactMap);//Log

if(prodPltformMap.size(>0))System.debug('====Case ID Blank
prodPltformMap===='+prodPltformMap);//Log
// Check the
Lookup table support level value exit for case

if(contactMap.size(>0) &&
contactMap.get(caseRec.ContactId).Support_Center__c!=null &&
contactMap.get(caseRec.ContactId).Parent_Account__c!=null &&
prodPltformMap.size(>0) && prodPltformMap.get(caseRec.Product__c)!
=null &&
prodPltformMap.get(caseRec.Product__c).Product_Attribute__r.Cadence_
Platform__c!=null &&
resLUTMap.get(contactMap.get(caseRec.ContactId).Support_Center__c+co
ntactMap.get(caseRec.ContactId).Parent_Account__c+prodPltformMap.get
(caseRec.Product__c).Product_Attribute__r.Cadence_Platform__c)!
=null)
{
//
Assign the Lookup table support level value to case Support_Level__c

```

```

field                                                                    if(!
caseRec.Ecommerce_Case__c)

if(resLUTMap.get(contactMap.get(caseRec.ContactId).Support_Center__c
+contactMap.get(caseRec.ContactId).Parent_Account__c+prodPltformMap.
get(caseRec.Product__c).Product_Attribute__r.Cadence_Platform__c)!
=null)caseRec.Support_Level__c=resLUTMap.get(contactMap.get(caseRec.
ContactId).Support_Center__c+contactMap.get(caseRec.ContactId).Paren
t_Account__c+prodPltformMap.get(caseRec.Product__c).Product_Attribut
e__r.Cadence_Platform__c).Support_Level__c;
                                                                    }//If ends
                                                                    }//Else If ends
                                                                    }//If ends
                                                                    }//For Ends
                                                                    }//If ends
}catch(exception e){System.debug('====Error Found
====='+e);}

} //Method ends

/**
 * @description #SFDC-4707 Assign Lookup Table Support Level
values for Cases from assignMultipleCases and onBeforeInsert method
only for Default Cases
 * @param caseRecData The case for whcih the assessment has
to be done
 * @param resLUTMap The collection of LUT tables fetched
 * @param caseListData the complete case list
 * @param prodPltformMap product and platform map
 * @param contactMap The collection of contacts
 */
Public static void LUTSupportLevelAssign(Case caseRecData,
Map<String,Lookup_Table__c> resLUTMap, List <Case> caseListData, Map
< String, Product_Class__c > prodPltformMap, Map < string>Contact>
contactMap){
    // Variable declarations
    List < Case > caseList = new List < Case > ();

    try{

if(resLUTMap.size()>0)System.debug('====resLUTMap===='+resLUTMap);/
/Log
        // Assign the single case or list of case
records to caseList variable
        if(caseRecData!=null &&
caseRecData.ContactId!=null)caseList.add(caseRecData);

if(caseListData.size()>0)caseList.addAll(caseListData);

        if(resLUTMap.size()>0){
            for (Case caseRec; caseList){

```



```

System.debug('====Case
Data===='+caseRec);//Log

if(CaseTriggerHandler_AC.recTypeMap.get(caseRec.RecordTypeId).trim()
.equalsIgnoreCase('Default') && caseRec.Case_Type__c != 'E2E')//
caseRec.Assignment_Status__c ==
CaseAssignmentEngine_AC.CHECK_ASSIGNMENT
{
// Check the Lookup
table support level value exit for case
if(caseRec.id!=null
&& caseRec.Contact.Support_Center__c!=null &&
caseRec.Contact.Parent_Account__c!=null &&
caseRec.Product__r.Product_Attribute__r.Cadence_Platform__c!=null)
{

System.debug('====Case ID
Found===='+caseRec.Contact.Support_Center__c+caseRec.Contact.Parent_
Account__c+caseRec.Product__r.Product_Attribute__r.Cadence_Platform_
_c);//Log
// Assign
the Lookup table support level value to case Support_Level__c field

if(resLUTMap.get(caseRec.Contact.Support_Center__c+caseRec.Contact.P
arent_Account__c+caseRec.Product__r.Product_Attribute__r.Cadence_Pla
tform__c)!
=null)caseRec.Support_Level__c=resLUTMap.get(caseRec.Contact.Support
_Center__c+caseRec.Contact.Parent_Account__c+caseRec.Product__r.Prod
uct_Attribute__r.Cadence_Platform__c).Support_Level__c;
} //If ends
else{

if(contactMap.size(>0))System.debug('====Case ID Blank
contactMap===='+contactMap);//Log

if(prodPltformMap.size(>0))System.debug('====Case ID Blank
prodPltformMap===='+prodPltformMap);//Log
// Check the
Lookup table support level value exit for case

if(contactMap.size(>0) &&
contactMap.get(caseRec.ContactId).Support_Center__c!=null &&
contactMap.get(caseRec.ContactId).Parent_Account__c!=null &&
prodPltformMap.size(>0) && prodPltformMap.get(caseRec.Product__c)!
=null &&
prodPltformMap.get(caseRec.Product__c).Product_Attribute__r.Cadence_
Platform__c!=null &&
resLUTMap.get(contactMap.get(caseRec.ContactId).Support_Center__c+co
ntactMap.get(caseRec.ContactId).Parent_Account__c+prodPltformMap.get
(caseRec.Product__c).Product_Attribute__r.Cadence_Platform__c)!
=null)

{
//

```

Assign the Lookup table support level value to case Support\_Level\_\_c field

```
if(resLUTMap.get(contactMap.get(caseRec.ContactId).Support_Center__c+contactMap.get(caseRec.ContactId).Parent_Account__c+prodPltformMap.get(caseRec.Product__c).Product_Attribute__r.Cadence_Platform__c)!=null)caseRec.Support_Level__c=resLUTMap.get(contactMap.get(caseRec.ContactId).Support_Center__c+contactMap.get(caseRec.ContactId).Parent_Account__c+prodPltformMap.get(caseRec.Product__c).Product_Attribute__r.Cadence_Platform__c).Support_Level__c;
                                                                    }//If ends
```

```
                                                                    }//Else If ends
```

```
                                                                    }//If ends
```

```
                                                                    }//For Ends
```

```
                                                                    }//If ends
```

```
                                                                    }catch(exception e){System.debug('====Error Found
===='+e);}
                                                                    }
```

```
                                                                    }//Method ends
```

```
/**
 * @description SFDC-4897 Get the Lookup Table Support Level
 * values for Cases from onBeforeInsert method only for Default Cases
 * @param caseRec The current contact record
 * @param resLUTMap lookup table contents
 * @param prodPltformMap product class information
 * @param contactMap contact information
 * @return String the support type based on the configuration
 */
Public static String evaluatedSupportLevel(Case caseRec,
Map<String,Lookup_Table__c> resLUTMap,Map < String, Product_Class__c
> prodPltformMap, Map < string,Contact> contactMap){
    String retSupportLevel='';
```

```
        if(caseRec.ContactId!=null && caseRec.Product__c!=null && contactMap.size()>0 &&
contactMap.get(caseRec.ContactId).Support_Center__c!=null &&
contactMap.get(caseRec.ContactId).Parent_Account__c!=null &&
        prodPltformMap.size()>0 &&
prodPltformMap.get(caseRec.Product__c)!=null &&
prodPltformMap.get(caseRec.Product__c).Product_Attribute__r.Cadence_Platform__c!=null &&
```

```
resLUTMap.get(contactMap.get(caseRec.ContactId).Support_Center__c+contactMap.get(caseRec.ContactId).Parent_Account__c+prodPltformMap.get(caseRec.Product__c).Product_Attribute__r.Cadence_Platform__c)!=null)
    {
```

```
        //Get the Lookup table support level value
```

```
retSupportLevel=resLUTMap.get(contactMap.get(caseRec.ContactId).Support_Center__c+contactMap.get(caseRec.ContactId).Parent_Account__c+prodPltformMap.get(caseRec.Product__c).Product_Attribute__r.Cadence_Platform__c).Support_Level__c;
```

```

odPltformMap.get(caseRec.Product__c).Product_Attribute__r.Cadence_Pl
atform__c).Support_Level__c;
    }//If ends
    return retSupportLevel;

```

```

} //Method ends

```

```

/**
 * @description Evaluate the support level that has been
 * specifically created for ecomm cases and retur the same
 * @param caseRec The current contact record
 * @param resLUTMap lookup table contents
 * @param prodPltformMap product class information
 * @param contactMap contact information
 * @param ent selected entitlement information
 * @return String the support type based on the configuration
 */

```

```

    Public static String evaluatedEcomSupportLevel(Case caseRec,
    Map<String,Lookup_Table__c> resLUTMap,Map < String, Product_Class__c
    > prodPltformMap, Map < string,Contact> contactMap, Entitlement ent)
    {

```

```

        String retSupportLevel='';
        System.debug('evaluatedEcomSupportLevel ECOM
'+caseRec.Product__c);

```

```

        if(caseRec.ContactId!=null && caseRec.Product__c!
        =null && contactMap.size()>0 &&
        contactMap.get(caseRec.ContactId).Support_Level__c!=null &&
        contactMap.get(caseRec.ContactId).Parent_Account__c!=null &&
        caseRec.Product__c!=null){
            System.debug(' inside if
evaluatedEcomSupportLevel ECOM ');

```

```

        if(resLUTMap.get(contactMap.get(caseRec.ContactId).Support_Level__c+
        contactMap.get(caseRec.ContactId).Customer_Type__c+ent.Product_Statu
        s__c+ent.Maint_Status__c+caseRec.Product__c)!=null)
        {

```

```

            //Get the Lookup table
            support level value

```

```

            retSupportLevel=resLUTMap.get(contactMap.get(caseRec.ContactId).Supp
            ort_Level__c+contactMap.get(caseRec.ContactId).Customer_Type__c+ent.
            Product_Status__c+ent.Maint_Status__c+caseRec.Product__c).Support_Le
            vel__c;

```

```

        }
        else

```

```

        if(resLUTMap.get(contactMap.get(caseRec.ContactId).Support_Level__c+
        contactMap.get(caseRec.ContactId).Customer_Type__c+ent.Product_Statu
        s__c+ent.Maint_Status__c)!=null){

```

```

            retSupportLevel=resLUTMap.get(contactMap.get(caseRec.ContactId).Supp

```

```
    ort_Level__c+contactMap.get(caseRec.ContactId).Customer_Type__c+ent.  
    Product_Status__c+ent.Maint_Status__c).Support_Level__c;  
        }  
    }  
    return retSupportLevel;  
}  
  
}  
/* CodeOptimization Marker */
```