

The screenshot shows a Microsoft Visual Studio interface with the following details:

- Menu Bar:** File, Edit, View, Project, Build, Debug, Team, Tools, Test, Analyze, Window, Help.
- Toolbars:** Standard, Debug, Task List, Solution Explorer, Toolbox, Properties, Task List, Status Bar.
- Code Editor:** The main window displays the code for **ACOController.cs**. The code is as follows:

```
1  using System;
2  using System.Collections.Generic;
3  using System.Data.Entity.Core.Objects;
4  using System.Linq;
5  using System.Net;
6  using System.Net.Http;
7  using System.Web;
8  using System.Web.Http;
9  using UQP.Models;
10 using UQP.Utility;
11
12 namespace UQP.Controllers
13 {
14
15     /// <summary>
16     /// This controller handles the request for ACO dashboard and ACO members listing
17     /// </summary>
18     [Authorize]
19     public class ACOController : ApiController
20     {
21         static int finalcount;
22
23
24         /// <summary>
25         /// Get the ACO data card
26         /// </summary>
27         /// <param name="objdate"></param>
28         /// <returns></returns>
29         [HttpPost]
30         [Route("api/ACO/dashboard/datacard")]
31     }
}
100 %
```

Edit View Project Build Debug Team Tools Test Analyze Window Help

ACOController.cs Source Control Explorer ACOComment.cs ACO_MembersMart.cs BulkEmailController.cs ACOMeasureReport.cs Notification_UserSpecific.cs

UQP

```
30 [HttpPost]
31 [Route("api/ACO/dashboard/datacard")]
32     References
33     public HttpResponseMessage GetACODataCard(ACOInput objdate)
34     {
35         try
36         {
37             using (UQPEntities db = new UQPEntities(HttpContext.Current.Request.Headers.GetValues("IPA").FirstOrDefault()))
38             {
39
40                 //Get data count for each section of data card
41                 var datacounts = (from dashss in db.ACO_NewDashboardSuperset.AsNoTracking()
42                                   where dashss.Quarter == objdate.quarter && objdate.name == "executive"
43                                   select new
44                                   {
45                                       dashss.ProviderCount,
46                                       dashss.MemberCount,
47                                       dashss.GroupCount
48
49                               }).FirstOrDefault();
50
51                 ACO_NewDashboardSuperset dashobj = new ACO_NewDashboardSuperset
52                 {
53                     MemberCount = datacounts.MemberCount,
54                     ProviderCount = datacounts.ProviderCount,
55                     GroupCount = datacounts.GroupCount
56                 };
57
58
59                 return Request.CreateResponse(HttpStatusCode.OK, dashobj);
60
61             }
62         }
63     }
```

Output

Show output from: Source Control - Team Foundation

The server has been temporarily disconnected from source control because the server is unavailable. To attempt to reconnect to source control, close and then re-open this window.

```
5     dashss.ProviderCount,
6     dashss.MemberCount,
7     dashss.GroupCount
8
9   }).firstOrDefault();
10
11 ACO_NewDashboardSuperset dashobj = new ACO_NewDashboardSuperset
12 {
13     MemberCount = datacounts.MemberCount,
14     ProviderCount = datacounts.ProviderCount,
15     GroupCount = datacounts.GroupCount
16 };
17
18
19 return Request.CreateResponse(HttpStatusCode.OK, dashobj);
20 }
21
22 }
23
24 catch (Exception ex)
25 {
26     return Request.CreateResponse(HttpStatusCode.InternalServerError, ex.Message);
27 }
28
29
30 /// <summary>
```

put from: Source Control - Team Foundation

```
    /// <summary>
    /// <param name="objdate"></param>
    /// <returns></returns>
    [HttpPost]
    [Route("api/ACO/dashboard/membershipbycounty")]
    public HttpResponseMessage Getmembershipbycounty(ACOInput objdate)
    {
        try
        {
            using (UQPEntities db = new UQPEntities(HttpContext.Current.Request.Headers.GetValues("IPA").FirstOrDefault()))
            {

                ///Get data count for each section of data card
                var datacounts = (db.ACO_MembersMart.AsNoTracking().Where(x=> x.FolderDate == objdate.quarter &&
                    x.Date_of_Death==null)).ToList().GroupBy(e => e.County_Number).Select(result =>
                new
                {
                    County_Number = result.Key,
                    CountyName = result.Select(y => y.County_Name).FirstOrDefault(),
                    StateName = result.Select(y => y.State_Name).FirstOrDefault(),
                    membercount=result.Select(y=>y.HICN).Distinct().Count()

                }).OrderByDescending(x => x.membercount).Take(5).ToList();

                List<ACO_CountyMembership> countyMemberships = new List<ACO_CountyMembership>();
                foreach (var per in datacounts)
```

File Edit View Tools Test Analyze Window Help

Debug Any CPU

UQP

ACOController.cs ACOComment.cs ACO_MembersMail.cs BulkEmailController.cs ACOMeasureReport.cs finalcount

```
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134

    foreach (var per in datacounts)
    {
        ACO_CountyMembership acoobj = new ACO_CountyMembership
        {
            countyname = per.CountyName,
            countynumber = per.County_Number,
            statename = per.StateName,
            count = per.membercount
        };
        countyMemberships.Add(acoobj);
    }

    return Request.CreateResponse(HttpStatusCode.OK, countyMemberships);
}

catch (Exception ex)
{
    return Request.CreateResponse(HttpStatusCode.InternalServerError, ex.Message);
}
```

```
134
135     /// <summary>
136     /// Get the Gap count on dashboard by member name
137     /// </summary>
138     /// <param name="name"></param>
139     /// <param name="filter"></param>
140     /// <returns></returns>
141     [HttpPost]
142     [Route("api/ACO/dashboard/gapcount/{name}")]
143     public HttpResponseMessage GetGapCount(string name, ACO_Filter filter)
144     {
145         try
146         {
147
148             using (UQPEntities db = new UQPEntities(HttpContext.Current.Request.Headers.GetValues("IPA").FirstOrDefault()))
149             {
150                 db.Database.CommandTimeout = 240;
151
152                 var fromyear1 = filter.startdate.Value.Year;
153
154
155                 int[] ids1 = { 1, 2, 3, 4, 6, 7, 8, 9, 10, 5 };
156
157
158
159                 var query = db.ACMeasureReports.Where(e => (e.CalculatedMonth == filter.startdate)
160                     &&
161                     (ids1.Contains((int)e.MeasureId) && e.FromYear.Year == fromyear1
162                     ) && ((name == "GroupID" && e.GroupID != null) || (name == "ProviderNPI" && e.ProviderNPI != null && e.providername != null)))
163                     ) && ((name == "GroupID" ? e.groupName : e.ProviderNPI).Select(result =>
164                         .GroupBy(e => (name == "GroupID") ? e.groupName : e.ProviderNPI).Select(result =>
```

A screenshot of a Microsoft Visual Studio IDE interface. The menu bar includes File, Edit, View, Team, Tools, Test, Analyze, Window, Help, Debug (which is selected), Any CPU, and IIS Express (Google Chrome). The code editor displays C# code for a controller named UOP.Controllers.ACOController. The code performs a database query on ACOMeasureReports, filters by calculated month, measure ID, provider NPI, and group ID, then groups by keyname and sums values. It then orders the results and adds them to a list of AC0dashboardGapInfo objects before returning a response.

```
var query = db.ACOMeasureReports.Where(e => (e.CalculatedMonth == filter.startdate)
&& (ids1.Contains((int)e.MeasureId) && e.FromYear.Year == fromyear
) && ((name == "GroupID" && e.GroupID != null) || (name == "ProviderNPI" && e.ProviderNPI != null && e.providername != null)))
.GroupBy(e => (name == "GroupID") ? e.groupName : e.ProviderNPI).Select(result =>
new
{
    keyname = (name == "GroupID") ? result.Key
    : result.Select(x => x.providername).FirstOrDefault(),
    value = result.Select(x => x.gap).Sum()
}).OrderByDescending(x => x.value).Take(10).OrderBy(x => x.keyname).ToList();
List<AC0dashboardGapInfo> gapdashboardlist = new List<AC0dashboardGapInfo>();
if (query.Count() != 0)
{
    foreach (var value in query)
    {
        AC0dashboardGapInfo dashobj = new AC0dashboardGapInfo
        {
            keyname = value.keyname,
            value = value.value,
        };
        gapdashboardlist.Add(dashobj);
    }
}
return Request.CreateResponse(HttpStatusCode.OK, gapdashboardlist.OrderByDescending(x=>x.value));
}
catch (Exception ex)
```

Microsoft Visual Studio

File Project Build Debug Team Tools Test Analyze Window Help

Debug Any CPU IIS Express (Google Chrome)

ACOController.cs Source Control Explorer ACOComment.cs ACO_MembersMart.cs BulkEmailController.cs ACOMeasureReport.cs Notification_UseCase.cs

UQP

```
172     List<ACOdashboardGapInfo> gapashboardlist = new List<ACOdashboardGapInfo>();  
173     if (query.Count() != 0)  
174     {  
175         foreach (var value in query)  
176         {  
177             ACODashboardGapInfo dashobj = new ACODashboardGapInfo  
178             {  
179                 keyname = value.keyname,  
180                 value = value.value,  
181             };  
182             gapashboardlist.Add(dashobj);  
183         }  
184     }  
185     return Request.CreateResponse(HttpStatusCode.OK, gapashboardlist.OrderByDescending(x=>x.value));  
186 }  
187 catch (Exception ex)  
188 {  
189     return Request.CreateResponse(HttpStatusCode.InternalServerError, ex);  
190 }  
191 }
```

The screenshot shows a Microsoft Visual Studio interface with the following details:

- Title Bar:** Shows "UOP.CPO" and "IIS Express (Google Chrome)".
- Toolbars:** "Source Control Explorer", "ACOComment.cs", "ACO_MembersMart.cs", "BulkEmailController.cs", "ACOMeasureReport.cs", and "finalcount".
- Code Editor:** Displays the following C# code for the `ACOController` class:

```
[HttpGet]
[Route("api/ACO/Getgroups/{npi}")]
public HttpResponseMessage GetACOGroups(string npi)
{
    try
    {
        using (UQPEntities db = new UQPEntities(HttpContext.Current.Request.Headers.GetValues("IPA").FirstOrDefault()))
        {
            List<ACO_Groups> grouplist = new List<ACO_Groups>();
            dynamic acoreresult = null;
            if (npi == "null" || npi=="0")
            {
                acoreresult = (from aco in db.ACO_Groups
                               select aco).Distinct().ToList();
            }
            else
            {
                acoreresult = (from p in db.ACOProviders
                               join g in db.ACO_Groups on p.groupid equals g.GroupID
                               where p.C_IND_NPI == npi
                               select g).Distinct().ToList();
            }
            foreach (var per in acoreresult)
            {
                ACO_Groups group = new ACO_Groups
                {
                    GroupName = per.GroupName,
                    // Other properties...
                };
            }
        }
    }
}
```

The code implements an `HttpGet` action method named `GetACOGroups` that takes a string parameter `npi`. It uses Entity Framework to query the database. If `npi` is null or zero, it retrieves all distinct groups. Otherwise, it joins the `ACOProviders` and `ACO_Groups` tables to filter by the specified NPI and retrieve distinct groups.

```
Controller.cs  X Source Control Explorer  ACOComment.cs  -ACO_MemberDetail.cs  BulkEmailController.cs  View Test Results  UQP.Controllers.ACOController
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
787
788
789
789
790
791
792
793
794
795
796
797
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
888
889
889
890
891
892
893
894
895
896
897
898
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
978
979
979
980
981
982
983
984
985
986
987
987
988
988
989
989
990
991
992
993
994
995
996
997
998
999
```

Microsoft Visual Studio

File Edit View Project Build Debug Team Tools Test Analyze Window Help

ACOController.cs Source Control Explorer ACOComments.cs ACO_MembersMart.cs BulkEmailController.cs ACOMeasureReport.cs Notifications.aspx.cs

UQP

ACOController.cs

```
246
247     [HttpPost]
248     [Route("api/ACO/membership/")]
249     public HttpResponseMessage GetACO_MembersMart(ACOInput input)
250     {
251         try
252         {
253             using (UQPEntities db = new UQPEntities(HttpContext.Current.Request.Headers.GetValues("IPA").FirstOrDefault()))
254             {
255                 string dateofbirth = input.dob == null ? null : (input.dob.Value.ToString("MM/dd/yyyy"));
256                 dynamic acorestore = null;
257                 if (input.report == true)
258                 {
259
260                     acorestore = db.ACO_MembersMart.AsNoTracking().Where(aco => aco.Date_of_Death == null
261                         && aco.FolderDate == input.quarter &&
262                             (input.groupid == 0 || aco.GroupId == input.groupid)
263
264                         && (input.npi == null || aco.Individual_NPI == input.npi)
265                         && (input.dob == null || aco.Birth_Date == dateofbirth)
266                         && (input.membername == null || aco.First_Name.StartsWith(input.membername))
267                         || aco.Last_Name.StartsWith(input.membername))
268                         && (input.subsid == null || aco.HICN.StartsWith(input.subsid)))
269                     .GroupBy(category => category.HICN, (key, g) => g.OrderByDescending(e => e.FolderDate).FirstOrDefault());
270                     Select(grouped => new
271                     {
272                         grouped.HICN,
273                         grouped.First_Name,
274                         grouped.Last_Name,
275                         grouped.Sex,
276                         grouped.Birth_Date,
277                         grouped.Provider_Name
278                     });
279                 }
280             }
281         }
282     }
```

Output

ACOController.cs X Source Control Explorer ACOComment.cs ACO_MembersMart.cs BulkEmailController.cs ACOMeasureReport.cs Notification_UsingTemplate.cs

UQP

```
275 grouped.Sex,
276 grouped.Birth_Date,
277 grouped.Provider_Name
278
279 grouped.Group_TIN,
280 grouped.GroupId,
281 grouped.provider_id,
282 grouped.IsAssigned,
283 grouped.IsProspective
284 })
285     .Distinct().OrderBy(e => new { e.Last_Name, e.First_Name });
286 }
287 else
288 {
289     acoreresult = db.ACO_MembersMart.AsNoTracking().Where(aco => aco.Date_of_Death == null
290     && aco.FolderDate == input.quarter &&
291         (input.groupid == 0 || aco.GroupId == input.groupid)
292         && (input.npi == null || aco.Individual_NPI == input.npi)
293         && (input.dob == null || aco.Birth_Date == dateofbirth)
294     && (input.membername == null || aco.First_Name.StartsWith(input.membername)
295     || aco.Last_Name.StartsWith(input.membername))
296     && (input.subsid == null || aco.HICN.StartsWith(input.subsid)))
297     .GroupBy(category=> category.HICN, (key, g) => g.OrderByDescending(e => e.FolderDate).FirstOrDefault());
298     Select(grouped => new
299     {
300         grouped.HICN,
301         grouped.First_Name,
302         grouped.Last_Name,
303         grouped.Sex,
304         grouped.Birth_Date,
305         grouped.Provider_Name
306     });
307 }
```

```
ACOController.cs  Source Control Explorer  ACOComment.cs  ACO_MembersMart.cs  BulkEmailController.cs  ACOMeasureReport.cs  Notifier.cs
UQP  UQP.Controllers.ACOController  finalcount

305     grouped.Provider_Name
306
307     grouped.Group_TIN,
308
309
310     grouped.GroupId,
311     grouped.provider_id,
312     grouped.IsAssigned,
313     grouped.IsProspective
314 }
315     .Distinct().OrderBy(e => new { e.Last_Name, e.First_Name })
316     .Skip(input.pageSize * (input.pageNumber - 1))
317     .Take(input.pageSize).ToList();
318
319
320
321
322
323     finalcount = db.ACO_MembersMart.AsNoTracking().Where(aco => aco.Date_of_Death == null
324     && aco.FolderDate == input.quarter &&
325         (input.groupid == 0 || aco.GroupId == input.groupid)
326         && (input.npi == null || aco.Individual_NPI == input.npi)
327         && (input.dob == null || aco.Birth_Date == dateofbirth)
328     && (input.membername == null || aco.First_Name.StartsWith(input.membername)
329     || aco.Last_Name.StartsWith(input.membername))
330     && (input.subsid == null || aco.HICN.StartsWith(input.subsid)))
331     .GroupBy(category => category.HICN, (key, g) => g.OrderByDescending(e => e.FolderDate).FirstOrDefault());
332     Select(grouped => new
333     {
334         grouped.HICN,
335
336     }).Distinct().Count();
337 }
```

```
UQP ACO_MemberList.cs BulkEmailController.cs ACOMeasureReport.cs notifications.cs
335
336     }).Distinct().Count();
337 }
338
339
340
341
342     List<ACOMember> memberlist = new List<ACOMember>();
343
344     foreach (var per in acoresult)
345     {
346         ACOMember member = new ACOMember
347         {
348             First_Name = per.First_Name,
349             Last_Name = per.Last_Name,
350             HICN = per.HICN,
351             Sex = per.Sex,
352             Birth_Date = per.Birth_Date,
353             Provider_Name = per.Provider_Name == null ? "NA" : per.Provider_Name,
354             provider_id = per.provider_id,
355             Group_TIN = per.Group_TIN == null ? "NA" : per.Group_TIN,
356             GroupId = per.GroupId,
357             IsProspective = per.IsProspective,
358             IsAssigned = per.IsAssigned
359         };
360         memberlist.Add(member);
361     }
362
363     if (input.report != true)
364     {
365         [pagination]
366     }
367 }
```

```
if (input.report != true)
{
    #region pagination
    // Get's No of Rows Count
    int count = finalcount;
    //int count = lstmembermart.Count();

    // Parameter is passed from Query string if it is null then it default Value will be pageNumber:1
    int CurrentPage = input.pageNumber;

    // Parameter is passed from Query string if it is null then it default Value will be pageSize:20
    int PageSize = input.pageSize;

    // Display TotalCount to Records to User
    int TotalCount = count;

    // Calculating Totalpage by Dividing (No of Records / Pagesize)
    int TotalPages = (int)Math.Ceiling(count / (double)PageSize);

    // Returns List of Customer after applying Paging
    //
    var items = memberlist/* .Skip((CurrentPage - 1) * PageSize).Take(PageSize).ToList() */;

    // if CurrentPage is greater than 1 means it has previousPage
    var previousPage = CurrentPage > 1 ? "Yes" : "No";

    // if TotalPages is greater than CurrentPage means it has nextPage
    var nextPage = CurrentPage < TotalPages ? "Yes" : "No";

    // Object which we are going to send in header
    var paginationMetadata = new
{
```

```
var items = memberlist/* .Skip((CurrentPage - 1) * PageSize).Take(PageSize).ToList() */;

// if CurrentPage is greater than 1 means it has previousPage
var previousPage = currentPage > 1 ? "Yes" : "No";

// if TotalPages is greater than currentPage means it has nextPage
var nextPage = currentPage < TotalPages ? "Yes" : "No";

// Object which we are going to send in header
var paginationMetadata = new
{
    totalCount = totalCount,
    pageSize = PageSize,
    currentPage = currentPage,
    totalPages = TotalPages,
    previousPage,
    nextPage
};

// Setting Header
HttpContext.Current.Response.Headers.Add("Paging-Headers", Newtonsoft.Json.JsonConvert.SerializeObject(paginationMetadata));
//HttpContext.Current.Response.Headers.Add("Filename", zipFileName);
HttpContext.Current.Response.Headers.Add("Access-Control-Expose-Headers", "Paging-Headers");
return Request.CreateResponse(HttpStatusCode.OK, memberlist);
#endregion
}
else
{
    var rpoj = new ReportsController
    {
        REQUEST = new System.Net.Http.HttpRequestMessage()
    }
}
```

putfrom: Source Control - Team Foundation
tive solution has been temporarily disconnected from source control because the server is unavailable. To attempt to reconnect to source control, close and then re

Source Control Explorer ACOComments.cs ACO_MembersMart.cs BulkEmailController.cs ACOMeasureReport.cs Notification_UserSpecific.cs

UQP.Controllers.ACOController GetACO_MembersMart(ACOInput input)

```
        };
        // Setting Header
        HttpContext.Current.Response.Headers.Add("Paging-Headers", Newtonsoft.Json.JsonConvert.SerializeObject(paginationMetadata));
        //HttpContext.Current.Response.Headers.Add("Filename", zipFileName);
        HttpContext.Current.Response.Headers.Add("Access-Control-Expose-Headers", "Paging-Headers");
        return Request.CreateResponse(HttpStatusCode.OK, memberlist);
    #endregion
}
else
{
    var rpoj = new ReportsController
    {
        Request = new System.Net.Http.HttpRequestMessage(),
        Configuration = new HttpConfiguration()
    };
    return rpoj.GenerateNotifReport(null,null, null, null, null, null, "ACO Member", "excel", null, null, null, null, null,
        null, null, null, null, memberlist, null, null, null, null, null, null);
}
}
}
catch (Exception e)
{
    return Request.CreateResponse(e.Message);
}
}
}
}

[HttpPost]

```

Source Control - Team Foundation

The solution has been temporarily disconnected from source control because the server is unavailable. To attempt to reconnect to source control, close and then re-open the solution.

```
431
432 [HttpPost]
433 [Route("api/ACO/GetACOProviders/")]
434 public HttpResponseMessage GetACOProviders(ACOInput input)
435 {
436     try
437     {
438         using (UQPEntities db = new UQPEntities(HttpContext.Current.Request.Headers.GetValues("IPA").FirstOrDefault()))
439         {
440
441             //var acoresult = (from aco in db.ACOProviders join addr in db.ACOProviderAddresses
442             //on aco.C_IND_NPI equals addr.NPI
443             //where aco.C_IND_NPI!=null && addr.NPI!=null &&
444             //(input.groupid == 0 || aco.groupid == input.groupid)
445             //  && (input.name == null || aco.firstname.StartsWith(input.name)
446             //  || aco.lastname.StartsWith(input.name))
447             //select new { aco.firstname, aco.lastname, aco.C_IND_NPI,
448             //            aco.EMAIL, aco.CENTER_NAME, aco.GEN, aco.PCP_SPC,
449             //            aco.TIN, aco.TEAM_LEAD, aco.SPECIALTY, aco.PROF,
450             //            addr.Address, addr.city, addr.phone, addr.fax, addr.zip,
451             //            aco.GROUP_TIN
452             //        }).ToList();
453
454             dynamic acoresult = null;
455             int acocount = 0;
456             if (input.report == true)
457             {
458                 if (input.key == "dashboardprovider")
459                 {
460                     acoresult = (from aco in db.ACO_MembersMart
461
462                         where aco.Individual_NPI != null &&
```

The screenshot shows a Microsoft Visual Studio interface with the following details:

- Title Bar:** Displays "Debug Any CPU IIS Express (Google Chrome)".
- Toolbars:** Standard Visual Studio toolbars for file operations, search, and navigation.
- Status Bar:** Shows "Scanned with CamScanner".
- Code Editor:** Contains C# code for an ASP.NET controller. The code filters a database table named "ACO_MembersMart" based on input parameters and performs a LINQ query to select specific fields like first name, last name, and TIN.
- Solution Explorer:** Shows files like "ACOComment.cs", "ACO_MembersMart.cs", "BulkEmailController.cs", "ACOMeasureReport.cs", and "Notification_UserSpecific.cs".
- Task List:** Shows a task named "GetACO_MembersMart(ACOInput input)".

```
if (input.key == "dashboardprovider")
{
    acoreult = (from aco in db.ACO_MembersMart
                where aco.Individual_NPI != null &&
                      (input.npi_number == null || aco.Individual_NPI == input.npi_number) &&
                      (input.groupid == 0 || aco.GroupId == input.groupid)
                && (input.quarter == null || aco.FolderDate == input.quarter)
                && aco.Date_of_Death == null
                select new
                {
                    firstname = "",
                    lastname = aco.Provider_Name,
                    C_IND_NPI = aco.Individual_NPI,
                    // aco.EMAIL,
                    // aco.CENTER_NAME,
                    // aco.GEN,
                    // aco.PCP_SPC,
                    TIN = aco.ACO_Participant_TIN,
                    // aco.TEAM LEAD,
                    // aco.SPECIALTY,
                    // aco.PROF,
                    GROUP_TIN = aco.Group_TIN
                }).Distinct().OrderBy(x => x.lastname==null).ThenBy(x=>x.lastname).ThenBy(x=>x.firstname)
            .Skip(input.pageSize * (input.pageNumber - 1))
            .Take(input.pageSize).ToList();
}
```

```
        .Skip(input.pageSize * (input.pageNumber - 1))
        .Take(input.pageSize).ToList();
    }

    else
    {
        acoreSult = (from aco in db.ACOProviders
                     where aco.C_IND_NPI != null &&
                           (input.npi_number == null || aco.C_IND_NPI == input.npi_number) &&
                           (input.groupid == 0 || aco.groupid == input.groupid)
                         && (input.name == null || aco.firstname.Trim().StartsWith(input.name))
                         || aco.lastname.Trim().StartsWith(input.name))
                     //&& aco.ACO == input.ACO
                     select new
                     {
                         aco.firstname,
                         aco.lastname,
                         aco.C_IND_NPI,
                         // aco.EMAIL,
                         // aco.CENTER_NAME,
                         // aco.GEN,
                         // aco.PCP_SPC,
                         aco.TIN,
                         // aco.TEAM LEAD,
                         // aco.SPECIALTY,
                         // aco.PROF,
                         aco.GROUP_TIN,
                         aco.ACO
                     }).Distinct()
                     .OrderBy(x => new { x.lastname, x.firstname })
                     .Skip(input.pageSize * (input.pageNumber - 1))
```

er.cs X Source Control Explorer ACOComments.cs ACO_MembersMart.cs BulkEmailController.cs ACOMeasureReport.cs Notification_Use_Specific.cs

Debug Any CPU IIS Express (Google Chrome) UQP.Controllers.ACOController GetACO_MembersMart(ACOInput input)

```
        }
    }

    if (input.report != true)
    {
        if (input.key == "dashboardprovider")
        {
            acocount = (from aco in db.ACO_MembersMart
                        where aco.Individual_NPI != null &&
                            (input.npi_number == null || aco.Individual_NPI == input.npi_number) &&
                            (input.groupid == 0 || aco.GroupId == input.groupid)
                            && (input.quarter == null || aco.FolderDate == input.quarter)
                            && aco.Date_of_Death == null
                        select new
                        {
                            ...
                            aco.Individual_NPI
                        }).Distinct().Count();
        }
        else
        {
            acocount = (from aco in db.ACOProviders
                        where aco.C_IND_NPI != null &&
                            (input.npi_number == null || aco.C_IND_NPI == input.npi_number) &&
```

```
roller.cs + X | Source Control Explorer | ACOComment.cs | ACO_MembersMart.cs | BulkEmailController.cs | ACOMeasureReport.cs | Notification_UserSpecification.cs
UQP.Controllers.ACOController | GetACO_MembersMart(ACOInput input)

        where aco.C_IND_NPI != null &&
        (input.npi_number == null || aco.C_IND_NPI == input.npi_number) &&
        (input.groupid == 0 || aco.groupid == input.groupid)
        && (input.name == null || aco.firstname.Trim().StartsWith(input.name)
        || aco.lastname.Trim().StartsWith(input.name))
    select new
    {
        aco.C_IND_NPI
    }).Distinct()
    .Count();
}

}

List<acoprovders> provlist = new List<acoprovders>();

foreach (var per in acoreresult)
{
    acoprovders obj = new acoprovders();
    List<ACOproviderAddress> listaddress = new List<ACOproviderAddress>();
    string[] name = per.AssuranceFirstName.Split(' ');
    /////
    //reportobj.AssuranceFirstName = name[0];
    //reportobj.AssuranceLastName = name[1];
    //if (per.lastname!=null && input.key == "dashboardprovider" && per.lastname.Contains(","))
    if (per.lastname!=null && input.key == "dashboardprovider" && per.lastname.Contains(","))
    {
        string[] name = per.lastname.Split(',');
        obj.firstname = name[1];
    }
}
```

Output from: Source Control - Team Foundation | Disconnected from source control because the server is unavailable. To attempt to reconnect to source control, close and then re-open this window.

COController.cs X Source Control Explorer ACOComment.cs ACO_MembersMart.cs BulkEmailController.cs ACOMeasureReport.cs NotificationsController.cs

UQP

```
649
650     string[] name = per.lastname.Split(',');
651     obj.firstname = name[1];
652
653     obj.lastname = name[0];
654     ///reportobj.AssuranceLastName = name[1];
655 }
656 else
657 {
658     obj.firstname = per.firstname;
659     obj.lastname = per.lastname;
660 }
661 obj.GROUP_TIN = per.GROUP_TIN;
662 //obj.PROF =per.PROF;
663 //obj.PCP_SPC =per.PCP_SPC;
664 //obj.GEN =per.GEN;
665 obj.C_IND_NPI = per.C_IND_NPI;
666
667 obj.TIN = per.TIN;
668 //obj.CENTER_NAME =per.CENTER_NAME;
669 //obj.SPECIALTY =per.SPECIALTY;
670 //obj.TEAM_LEAD =per.TEAM_LEAD;
671 // obj.EMAIL = per.EMAIL;
672 obj.ACO = (per.ACO == null) ? null : per.ACO;
673
674
675 var acoproviderAddress = db.ACOProviderAddresses.Where(x => (x.NPI == obj.C_IND_NPI))
676 ).Select(x => x).ToList();
677
678 foreach (var pers in acoproviderAddress)
679 {
680     ACOProviderAddress addrobj = new ACOProviderAddress();
681 }
```

The screenshot shows a Microsoft Visual Studio interface with the following details:

- Project:** UQP
- File:** ACOController.cs
- Code View:** The code is displayed in a scrollable window.
- Toolbars:** Standard Visual Studio toolbars are visible at the top.
- Status Bar:** The status bar at the bottom right indicates "Scanned with CamScanner".

```
673
674
675
676     var acoproviderAddress = db.ACOProviderAddresses.Where(x => (x.NPI == obj.C_IND_NPI)
677     ).Select(x => x).ToList();
678
679     foreach (var pers in acoproviderAddress)
680     {
681         ACOProviderAddress addrobj = new ACOProviderAddress();
682         addrobj.Address = pers.Address;
683         addrobj.city = pers.city;
684         addrobj.fax = pers.fax;
685         addrobj.phone = pers.phone;
686         listaddress.Add(addrobj);
687     }
688     obj.addresslist = listaddress;
689
690
691
692     provlist.Add(obj);
693 }
694 if (input.report != true)
695 {
696     [pagination]
697     return Request.CreateResponse(HttpStatusCode.OK, items);
698 }
699 else
700 {
701     var rpoj = new ReportsController()
702 }
```

```
1 if (input.report != true)
2 {
3     //region pagination
4     // Get's No of Rows Count
5     int count = acount;
6     //int count = 1stmembermart.Count();
7
8     // Parameter is passed from Query string if it is null then it default Value will be pageNumber:1
9     int currentPage = input.pageNumber;
10
11    // Parameter is passed from Query string if it is null then it default Value will be pageSize:20
12    int PageSize = input.pageSize;
13
14    // Display TotalCount to Records to User
15    int TotalCount = count;
16
17    // Calculating Totalpage by Dividing (No of Records / Pagesize)
18    int TotalPages = (int) Math.Ceiling(count / (double)PageSize);
19
20    // Returns List of Customer after applying Paging
21    var items = provlist;
22
23    // if CurrentPage is greater than 1 means it has previousPage
24    var previousPage = currentPage > 1 ? "Yes" : "No";
25
26    // if TotalPages is greater than currentPage means it has nextPage
27    var nextPage = currentPage < TotalPages ? "Yes" : "No";
28
29    // Object which we are going to send in header
30    var paginationMetadata = new
31    {
32        totalCount = TotalCount,
```

output from: Source Control - Team Foundation
active solution has been temporarily disconnected from source control because the server is unavailable. To attempt to reconnect to source control, close and then

```
int totalCount = count;

// Calculating Totalpage by Dividing (No of Records / Pagesize)
int TotalPages = (int)Math.Ceiling(count / (double)pageSize);

// Returns List of Customer after applying Paging
var items = provlist;

// if CurrentPage is greater than 1 means it has previousPage
var previousPage = currentPage > 1 ? "Yes" : "No";

// if TotalPages is greater than CurrentPage means it has nextPage
var nextPage = currentPage < TotalPages ? "Yes" : "No";

// Object which we are going to send in header
var paginationMetadata = new
{
    totalCount = TotalCount,
    pageSize = PageSize,
    currentPage = currentPage,
    totalPages = TotalPages,
    previousPage,
    nextPage
};

// Setting Header
HttpContext.Current.Response.Headers.Add("Paging-Headers", Newtonsoft.Json.JsonConvert.SerializeObject(paginationMetadata));
//HttpContext.Current.Response.Headers.Add("Filename", zipFileName);
//HttpContext.Current.Response.Headers.Add("Access-Control-Expose-Headers", "Paging-Headers");
#endregion

return Request.CreateResponse(HttpStatusCode.OK, items);

```

Show output from: Source Control - Team Foundation
The active solution has been temporarily disconnected from source control because the server is unavailable. To attempt to reconnect to source control, close and then re-open this solution.

SOURCE CONTROL EXPLORER ACOComment.cs ACO_MembersMart.cs BulkEmailController.cs ACOMeasureReport.cs Notification_UserSpecific.cs

UQP.Controllers.ACOController GetACOProviders(ACOInput input)

```
// Setting Header

HttpContext.Current.Response.Headers.Add("Paging-Headers", Newtonsoft.Json.JsonConvert.SerializeObject(paginationMetadata));
//HttpContext.Current.Response.Headers.Add("Filename", zipFileName);
HttpContext.Current.Response.Headers.Add("Access-Control-Expose-Headers", "Paging-Headers");
#endregion

return Request.CreateResponse(HttpStatusCode.OK, items);
}
else
{
    var rpoj = new ReportsController
    {
        Request = new System.Net.Http.HttpRequestMessage(),
        Configuration = new HttpConfiguration()
    };
    return rpoj.GenerateNotifReport(null,null, null, null, null, null, "ACO Provider", "excel", null, null, null, null,
        null, null, null, null, null, null, provlist, null, null,null, null, null);
}
}
catch (Exception e)
{
    return Request.CreateResponse(e.Message);
}
}
```

The status bar at the bottom of the IDE window displays the message: "Source control is unavailable. To attempt to reconnect to source control, close and then re-open the solution."

Source Control Explorer ACOComment.cs ACO_MembersMart.cs BulkEmailController.cs ACOMeasureReport.cs Notification_UserSpecific.cs
UQP.Controllers.ACOController GetACOProviders(ACOInput input)

```
[HttpPost]
[Route("api/ACO/GroupMgmt")]
public HttpResponseMessage GroupMgmt(ACOInput input)
{
    try
    {
        using (UQPEntities db = new UQPEntities(HttpContext.Current.Request.Headers.GetValues("IPA").FirstOrDefault()))
        {

            var query = db.ACOProviders.Join(db.ACO_Groups, gr => gr.groupid, pr => pr.GroupID,
                (gr, pr) => new { gr, pr })
                .Where(x => ((input.status == null) || x.pr.status == input.status)
                && (x.gr.C_IND_NPI != null)
                && (input.groupid == 0 || x.gr.groupid == input.groupid) &&
                (input.name == null || x.gr.GROUP_TIN.StartsWith(input.name)))
                .GroupBy(x => new { x.gr.groupid }).Select(r =>
                new
                {
                    providercount = r.Select(x => x.gr.C_IND_NPI).Distinct().Count(),
                    groupname = r.Select(x => x.gr.GROUP_TIN).FirstOrDefault(),
                    tinn = r.Select(x => x.gr.TIN).FirstOrDefault(),
                    groupid = r.Select(x => x.gr.groupid).FirstOrDefault(),
                    status = r.Select(x => x.pr.status).FirstOrDefault()
                }).ToList();
        }
    }
}
```

from Source Control - Team Foundation
The solution has been temporarily disconnected from source control because the server is unavailable. To attempt to reconnect to source control, close and then re-open


```
    ACO_MembersMart.cs      BulkEmailController.cs      ACOMeasureReport.cs      NotificationManager.cs
  ↳ UQP.Controllers.ACOController      ↳ GetACOProviders(ACOInput input)

if (Input.Key == "dashboardprovider")
{
    //query = (from result in query
    //          join mm in db.ACO_MembersMart
    //            on result.groupid equals mm.GroupId
    //          where mm.FolderDate == input.quarter
    //          select new { result.groupid, result.groupname, result.tinno, result.providercount, result.status }).Distinct()
    //          .ToList();

    query = (from result in query
              join mm in db.ACO_MembersMart
                on result.groupid equals mm.GroupId
              where mm.FolderDate == input.quarter
              select result).Distinct()
              .ToList();
}

List<ACOgroupMgmt> grouplist = new List<ACOgroupMgmt>();

foreach (var per in query)
{
    ACOgroupMgmt obj = new ACOgroupMgmt();

    obj.providercount = per.providercount;
    obj.groupname = per.groupname;
    obj.tinno = per.tinno;
    obj.status = per.status;
    obj.groupid = Convert.ToInt32(per.groupid);
    grouplist.Add(obj);
}
```

from Source Control - Team Foundation

```
UQP.Controllers.ACOController.cs
UQP.Controllers.ACOController
obj.Status = per.Status;
obj.GroupId = Convert.ToInt32(per.GroupId);
groupList.Add(obj);

} if (input.Report != true)
{
    #region Pagination
    // Get's No of Rows Count
    int count = groupList.Count();
    //int count = lstMemberList.Count();

    // Parameter is passed from Query string if it is null then it default Value will be pageNumber:1
    int currentPage = input.pageNumber;

    // Parameter is passed from Query string if it is null then it default Value will be pageSize:20
    int pageSize = input.pageSize;

    // Display TotalCount to Records to User
    int totalCount = count;

    // Calculating Totalpage by Dividing (No of Records / Pagesize)
    int totalPages = (int)Math.Ceiling(count / (double)pageSize);

    // Returns List of Customer after applying Paging
    var items = groupList.Skip((currentPage - 1) * pageSize).Take(pageSize).ToList();

    // if currentPage is greater than 1 means it has previousPage
    var previousPage = currentPage > 1 ? "Yes" : "No";

    // if TotalPages is greater than currentPage means it has nextPage
    var nextPage = currentPage < totalPages ? "Yes" : "No";

    // Object which we are going to send in header
}
```

from Source Control - Team Foundation
The solution has been temporarily disconnected from source control because the server is unavailable. To attempt to reconnect to source control, close and then re-open

```
UQP.Controllers.ACOController          ACOMeasureReports          Application Insights
var previousPage = currentPage > 1 ? "Yes" : "No";

// if TotalPages is greater than CurrentPage means it has nextPage
var nextPage = currentPage < TotalPages ? "Yes" : "No";

// Object which we are going to send in header
var paginationMetadata = new
{
    totalCount = TotalCount,
    pageSize = PageSize,
    currentPage = currentPage,
    totalPages = TotalPages,
    previousPage,
    nextPage
};

// Setting Header
HttpContext.Current.Response.Headers.Add("Paging-Headers", Newtonsoft.Json.JsonConvert.SerializeObject(paginationMetadata));
//HttpContext.Current.Response.Headers.Add("Filename", zipFileName);
HttpContext.Current.Response.Headers.Add("Access-Control-Expose-Headers", "Paging-Headers");
#endregion

return Request.CreateResponse(HttpStatusCode.OK, items);
}
else
{
    var rpoj = new ReportsController
    {
        Request = new System.Net.Http.HttpRequestMessage(),
        Configuration = new HttpConfiguration()
    };
}
```

... attempt to reconnect to source control, close and then re-open

The screenshot shows a Microsoft Visual Studio interface with the following details:

- Menu Bar:** Project, Build, Debug, Team, Tools, Test, Analyze, Window, Help.
- Toolbox:** Standard icons for file operations like Open, Save, Print, etc.
- Task List:** A sidebar on the left showing a tree structure of files and folders.
- Code Editor:** The main window displays C# code for a controller named ACOLinked.cs. The code handles requests for generating reports and updating group status.
- Toolbars:** Standard Windows-style toolbars for navigating between windows and performing common tasks.

```
[HttpPost]
[Route("api/ACO/UpdateGroupStatus/")]
public HttpResponseMessage UpdateGroupStatus(ACOInput input)
```

```
    }
}

[HttpPost]
[Route("api/ACO/GenerateReport/")]
public HttpResponseMessage GenerateReport(ACOInput input)
{
    try
    {
        if (input != null)
        {
            if (input.type == "Group")
            {
                var rpoj = new ReportsController();
                var items = rpoj.GenerateNotifReport(null, null, null, null, null, null, "ACO Groups", "excel", null,
                    null, null, null, null, null, null, null, null, null, null, null, null, null, null, null, null, null);
                return Request.CreateResponse(HttpStatusCode.OK, items);
            }
            else
            {
                var rpoj = new ReportsController();
                {
                    Request = new System.Net.Http.HttpRequestMessage(),
                    Configuration = new HttpConfiguration();
                };

                return rpoj.GenerateNotifReport(null, null, null, null, null, null, "ACO Groups", "excel", null,
                    null, null, null, null, null, null, null, null, null, null, null, null, null, null, null, null);
            }
        }
        catch (Exception e)
        {
            return Request.CreateResponse(e.Message);
        }
    }
}

[HttpGet]
[Route("api/ACO/GetGroupList/")]
public HttpResponseMessage GetGroupList()
{
    try
    {
        var rpoj = new ReportsController();
        var grouplist = rpoj.GetGroupList();
        var response = Request.CreateResponse(HttpStatusCode.OK, grouplist);
        response.Headers.Add("Access-Control-Expose-Headers", "Paging-headers");
        return response;
    }
    catch (Exception e)
    {
        return Request.CreateResponse(e.Message);
    }
}
```

UQPController.cs Source Control Explorer ACOComments.cs ACO_MembersMart.cs BulkEmailController.cs ACOMeasureReport.cs NotificationList.aspx

UQP.Controllers.ACOController GroupMgmt(ACOInput input)

```
897
898
899
900
901 [HttpPost]
902 [Route("api/ACO/UpdateGroupStatus/")]
903 public HttpResponseMessage UpdateGroupStatus(ACOInput input)
904 {
905     try
906     {
907         using (UQPEntities db = new UQPEntities(HttpContext.Current.Request.Headers.GetValues("IPA").FirstOrDefault()))
908         {
909             var groupid = db.ACO_Groups.Where(x => x.GroupID == input.groupid).FirstOrDefault();
910
911             if (groupid != null)
912             {
913                 groupid.status = input.status;
914             }
915         }
916         db.SaveChanges();
917
918
919
920
921
922
923         return Request.CreateResponse(HttpStatusCode.OK, groupid);
924     }
925 }
926 catch (Exception e)
927 {
928 }
```

View Project Build Debug Team Tools Test Analyze Window Help

Controller.cs* Source Control Explorer ACOComment.cs ACO_MembersMart.cs BulkEmailController.cs ACOMeasureReport.cs GroupMgmt(ACOInput)

```
009
010     if (groupid != null)
011     {
012
013         groupid.status = input.status;
014
015     }
016     db.SaveChanges();
017
018
019
020
021
022
023
024         return Request.CreateResponse(HttpStatusCode.OK, groupid);
025     }
026 }
027 catch (Exception e)
028 {
029     return Request.CreateResponse(e.Message);
030 }
031
032 }
033
034 [HttpPost]
035 [Route("api/ACO/GetMembers/{type}")]
```

```
controller.cs*  X Source Control Explorer      ACOComment.cs      ACO_MembersMart.cs      BulkEmailController.cs      ACOMeasureReport.cs      Notification_UserSpecific.cs
P UQP.Controllers.ACOController      GroupMgmt(ACOInput input)

33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
```

```
UQP.Controllers.ACOController ACOMeasureReport.cs Notification_UserSpecific.cs
getdenominatorlist(ACO_Filter filter, string type)

if (filter.report == false)
{
    query = (from d in db.ACO_Measures_Members
              where
              (
d.CLM_FROM_DT <= filter.enddate
&& intersect1.Contains((int)d.MeasureId) && d.fromyear.Value.Year == fromyear1
&&
(filter.dob == null || d.DOB == filter.dob) &&
(filter.subscribername == null || (d.SubscriberFirstName.StartsWith(filter.subscribername) ||
d.SubscriberLastName.StartsWith(filter.subscribername) ||
(d.SubscriberLastName + "," + d.SubscriberFirstName).StartsWith(filter.subscribername)))
&&
(filter.hicn == null|| filter.hicn == "" || d.BENE_HIC_NUM.StartsWith( filter.hicn))
&& (filter.npi == null || filter.npi == "" || d.ProviderNPI == filter.npi)
&& (filter.groupid == null|| d.Group_ID == filter.groupid)
&& ((type == "Gaps") ? d.code_type == "D" : (type == "Numerator") ? d.code_type == "N" : (type == "Exclusion") ?
d.code_type == "E" : (d.code_type == "N" || d.code_type == "D"))
              )
    select new
    {
        d.BENE_HIC_NUM,
        d.SubscriberFirstName,
        d.SubscriberMidName,
        d.SubscriberLastName,
        d.DOB,
        d.Gender,
        d.MeasureName,
        d.MeasureId,
        d.ProviderNPI,
        d.ProviderFirstName,
        ...
    }
}
```

```
        d.MeasureId,
        d.ProviderNPI,
        d.ProviderFirstName,
        d.ProviderMidName,
        d.ProviderLastName,
        d.Source_TypeUQP
    }).Distinct().OrderBy(e => new { e.SubscriberLastName, e.SubscriberFirstName })
    .Skip(filter.pageSize * (filter.pageNumber - 1)).Take(filter.pageSize).ToList();

totalcount = (from d in db.ACO_Measures_Members where
    (
        d.CLM_FROM_DT <= filter.enddate
        && intersect1.Contains((int)d.MeasureId) && d.fromyear.Value.Year == fromyear1
        &&
        (filter.dob == null || d.DOB == filter.dob) &&
        (filter.subscribername == null || (d.SubscriberFirstName.StartsWith(filter.subscribername) ||
        d.SubscriberLastName.StartsWith(filter.subscribername)) ||
        (d.SubscriberLastName + "," + d.SubscriberFirstName).StartsWith(filter.subscribername)))
        &&
        (filter.hicn == null || filter.hicn == "" || d.BENE_HIC_NUM.StartsWith(filter.hicn))
        && (filter.npi == null || filter.npi == "" || d.ProviderNPI == filter.npi)
        && (filter.groupid == null || d.Group_ID == filter.groupid)
        && ((type == "Gaps") ? d.code_type == "D" : (type == "Numerator") ? d.code_type == "N" : (type == "Exclusion") ? d.code_type == "E" :
        (d.code_type == "N" || d.code_type == "D"))
    )
    select new
    {
        d.BENE_HIC_NUM,
        d.MeasureId,
```

from Source Control - Team Foundation
live solution has been temporarily disconnected from source control because the server is unavailable. To attempt to reconnect to source control, close and then re

```
d.MeasureId,  
}).Distinct().Count();  
}  
else if (filter.report == true)  
{  
    query = (from d in db.ACO_Measures_Members  
    where  
    (  
        d.CLM_FROM_DT <= filter.enddate &&  
        intersect1.Contains((int)d.MeasureId) && d.fromyear.Value.Year == fromyear1  
        &&  
        (filter.dob == null || d.DOB == filter.dob) &&  
        (filter.subscribername == null || (d.SubscriberFirstName.StartsWith(filter.subscribername) ||  
        d.SubscriberLastName.StartsWith(filter.subscribername) ||  
        (d.SubscriberLastName + "," + d.SubscriberFirstName).StartsWith(filter.subscribername)))  
        &&  
        (filter.hicn == null || d.BENE_HIC_NUM == filter.hicn)  
        && (filter.npi == null || filter.npi == "" || d.ProviderNPI == filter.npi)  
        && (filter.groupid == null || d.Group_ID == filter.groupid)  
        && ((type == "Gaps") ? d.code_type == "D" : (type == "Numerator") ? d.code_type == "N" : (type == "Exclusion") ?  
        d.code_type == "E" : d.code_type == "N" || d.code_type == "D")  
    )  
    select new  
    {  
        d.BENE_HIC_NUM,  
        d.SubscriberFirstName,  
        ...  
    };  
}
```

put from: Source Control - Team Foundation
The solution has been temporarily disconnected from source control because the server is unavailable. To attempt to reconnect to source control, close

```
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
988
989
989
990
991
992
993
994
995
996
997
998
999
```

```
1069 providerNPI = value.ProviderNPI,  
1070 hicn = value.BENE_HIC_NUM,  
1071 sourcetype=value.Source_TypeUQP  
1072 };  
1073 memberlist.Add(member);  
1074 }  
1075  
1076 if (filter.report == false)  
1077 {  
1078     #region pagination  
1079     // Get's No of Rows Count  
1080     int count = totalcount;  
1081     //lstmembercount.Count();  
1082     // int count = 0;  
1083  
1084  
1085     // Parameter is passed from Query string if it is null then it default Value will be pageNumber:1  
1086     int CurrentPage = filter.pageNumber;  
1087  
1088     // Parameter is passed from Query string if it is null then it default value will be pageSize:20  
1089     int PageSize = filter.pageSize;  
1090  
1091     // Display TotalCount to Records to User  
1092     int TotalCount = count;  
1093  
1094     // Calculating Totalpage by Dividing (No of Records / Pagesize)  
1095     int TotalPages = (int)Math.Ceiling(count / (double)PageSize);  
1096  
1097     // Returns List of Customer after applying Paging  
1098     var items = memberlist;//.Skip((CurrentPage - 1) * PageSize).Take(PageSize).ToList();
```

100 %

Output

Show output from: Source Control - Team Foundation
The active solution has been temporarily disconnected from source control because the server is unavailable. To attempt to reconnect to source control, close and

ACOController.cs* X Source Control Explorer ACOComment.cs ACO_MembersMart.cs BulkEmailController.cs ACOMeasureReport.cs Notification_UseType.cs

UQP UQP.Controllers.ACOController getdenominatorlist(ACO_Filter filter, string type)

```
1097 // Returns List of Customer after applying Paging
1098 var items = memberlist;//.Skip((CurrentPage - 1) * PageSize).Take(PageSize).ToList();
1099
1100 // if CurrentPage is greater than 1 means it has previousPage
1101 var previousPage = CurrentPage > 1 ? "Yes" : "No";
1102
1103 // if TotalPages is greater than CurrentPage means it has nextPage
1104 var nextPage = CurrentPage < TotalPages ? "Yes" : "No";
1105
1106 // Object which we are going to send in header
1107 var paginationMetadata = new
1108 {
1109     totalCount = TotalCount,
1110     pageSize = PageSize,
1111     currentPage = CurrentPage,
1112     totalPages = TotalPages,
1113     previousPage,
1114     nextPage
1115 };
1116
1117 // Setting Header
1118
1119 System.Web.HttpContext.Current.Response.Headers.Add("Paging-Headers", Newtonsoft.Json.JsonConvert.SerializeObject(paginationMetadata));
1120 //HttpContext.Current.Response.Headers.Add("Filename", zipFileName);
1121 HttpContext.Current.Response.Headers.Add("Access-Control-Expose-Headers", "Paging-Headers");
1122 return Request.CreateResponse(HttpStatusCode.OK, items);
1123 #endregion
1124 }
1125 else
1126 {
1127     var rpoj = new ReportsController
1128     {
```

100 % 4

Output

Show output from: Source Control - Team Foundation

The active solution has been temporarily disconnected from source control because the server is unavailable. To attempt to reconnect to source control, close and then

```
// Setting Header
.
.
.
System.Web.HttpContext.Current.Response.Headers.Add("Paging-Headers", Newtonsoft.Json.JsonConvert.SerializeObject(paginationMetadata));
//HttpContext.Current.Response.Headers.Add("Filename", zipFileName);
HttpContext.Current.Response.Headers.Add("Access-Control-Expose-Headers", "Paging-Headers");
return Request.CreateResponse(HttpStatusCode.OK, items);
#endregion
}
else
{
    var rpoj = new ReportsController
    {
        Request = new System.Net.Http.HttpRequestMessage(),
        Configuration = new HttpConfiguration()
    };
    return rpoj.GenerateNotifReport(null, null, null, null, null, null, "ACO " + type, "excel", null,
        null, null, null, null, null, null, null, null, null, memberlist, null, null);
}
}
}
catch (Exception ex)
{
    return Request.CreateResponse(HttpStatusCode.OK, ex);
}
}
```

from Source Control - Team Foundation
solution has been temporarily disconnected from source control because the server is unavailable. To attempt to reconnect to source control, close and then re-