I first check that all users who have interacted with the campaign have downloaded profiles stored against the syndication key. To do this I use the following (you must know the community id and the syndication key). Be aware some communities do not return any data but this should be handled as Unknowns.

DECLARE @commid int

DECLARE @syndid nvarchar(50)

SET @commid = 1

SET @syndid = 'dnuk.ds.lin.de'

SELECT COUNT(DISTINCT(b.GlobalUserId))

FROM Interaction a

INNER JOIN InteractionSession b ON a.sessionId = b.sessionId

INNER JOIN GlobalUserIds c ON b.GlobalUserId = c.GlobalUserId

INNER JOIN SyndicatedSiteCommunities d ON a.syndicationId = d.syndicationId AND c.communityId = d.communityId AND a.interactionDate >= d.trackingcleardate

INNER JOIN medical01.dbo.GlobalUserIds e ON b.GlobalUserId = e.GlobalUserId

WHERE a.syndicationId = @syndid AND c.communityid = @commid

AND b.GlobalUserId NOT IN (SELECT b.GlobalUserId

FROM Interaction a

INNER JOIN InteractionSession b ON a.sessionId = b.sessionId

INNER JOIN GlobalUserIds c ON b.GlobalUserId = c.GlobalUserId

INNER JOIN SyndicatedSiteCommunities d ON a.syndicationId = d.syndicationId AND c.communityId = d.communityId AND a.interactionDate >= d.trackingcleardate

INNER JOIN CommunityUserProfile e ON e.GlobalUserId = b.GlobalUserId

INNER JOIN dbo.CommunityProfileDictionary f ON e.DictionaryName = f.DictionaryName AND e.KeyName = f.KeyName

WHERE a.syndicationId = @syndid AND c.communityid = @commid AND e.SyndicationId = @syndid AND f.DictionaryName = 'specialty' AND f.CommunityId = @commid)

If the returned value is greater than 0 then there are profiles that need downloading. To download the profiles change the select in the above query to be

SELECT DISTINCT(b.GlobalUserId), e.CommunityUserKey

This will give you the GlobalUserId and the CommunityUserKey. You can then use the CommunityUserKey to request the profile download by hitting the following endpoint in postman (change {communityId} with the community id used above:

[https://www.doctors.net.uk/jsapi/v1.2/clinaccess/Profile.svc/{communityId}/user](https://www.doctors.net.uk/jsapi/v1.2/clinaccess/Profile.svc/%7bcommunityId%7d/user)

This endpoint is a post which takes the following json body (change the {CommunityUserKey} with the value from the query above and the {SyndicationKey} with the value used above):

{"data":"{\"userKey\":\"{CommunityUserKey}\", \"syndicationKey\": \"{SyndicationKey}\"}"}

The response from this endpoint is either true or false. True is the data was returned successfully, false means the data wasn’t returned successfully. If you want more detail you can look at the logs using the following query. Note there are no logs for community 1 as most of these would be forum or dnuk home instead of clinaccess campaigns.

SELECT \* FROM ProfileRequestLog ORDER BY requestTime DESC

Once you have all the profile for all the users you can use the following query to mimic what is done in the clinaccess dashboard, this particular version is to get a breakdown of the users by specialty.

DECLARE @commid int

DECLARE @syndid nvarchar(50)

SET @commid = 6

SET @syndid = 'dnuk.ds.lin.de'

SELECT f.KeyValue, COUNT(DISTINCT(c.GlobalUserId))

FROM Interaction a

INNER JOIN InteractionSession b ON a.sessionId = b.sessionId

INNER JOIN GlobalUserIds c ON b.GlobalUserId = c.GlobalUserId

INNER JOIN SyndicatedSiteCommunities d ON a.syndicationId = d.syndicationId AND c.communityId = d.communityId AND a.interactionDate >= d.trackingcleardate

INNER JOIN CommunityUserProfile e ON e.GlobalUserId = b.GlobalUserId

INNER JOIN dbo.CommunityProfileDictionary f ON e.DictionaryName = f.DictionaryName AND e.KeyName = f.KeyName

WHERE a.syndicationId = @syndid AND c.communityid = @commid AND e.SyndicationId = @syndid AND f.CommunityId = @commid AND f.DictionaryName = 'specialty' AND a.url NOT LIKE '%client.doctors%'