### Table of Contents – CRM Integration

[1. CRM –Overview 2](#_Toc479241912)

[1.1 Different Entities in CRM 2](#_Toc479241913)

[1.2 CMR terminologies 2](#_Toc479241914)

[2. CIS (Contact Integration Service) 3](#_Toc479241915)

[3. CER (Contact Entity Record) 3](#_Toc479241916)

[4. FER (Field Entity record) 4](#_Toc479241917)

[5. Config Entries 4](#_Toc479241918)

[6. BAM (Batch Attachments Manager) 5](#_Toc479241919)

[6.1 Functionality: - 5](#_Toc479241920)

[6.2 Testing BAM:- 5](#_Toc479241921)

[7. BAM deploy: - 5](#_Toc479241922)

[8. E-Manager, WebResource 6](#_Toc479241923)

[8.1 E-Manager App functionality 6](#_Toc479241924)

[8.2 Web resources: - 6](#_Toc479241925)

[9. AD Explorer 6](#_Toc479241926)

[10. CRM Plug-Ins 7](#_Toc479241927)

[11. Plug-In Registration tool 7](#_Toc479241928)

[12. Code flow for Merge plug-in Functionality 8](#_Toc479241929)

[12.1 Assign, share and unshare contacts on CRM 9](#_Toc479241930)

[12.1.1 Assign contact 9](#_Toc479241931)

[12.1.2 Share contact 9](#_Toc479241932)

[12.1.3 Unshare contact 9](#_Toc479241933)

[12.1.4 Merge contact 9](#_Toc479241934)

[13. CDAS – CRMData Access Service 10](#_Toc479241935)

[14. CDAS- Code flow 10](#_Toc479241936)

[15. Launching various Apps from CRM 11](#_Toc479241937)

# 

# CRM –Overview

MS CRM -Is a [customer relationship management](https://en.wikipedia.org/wiki/Customer_relationship_management) software package which focuses mainly on Sales, Marketing, and Service sectors. CRM is used for advertise and sale of insurance products to the interested contacts/clients, provide service to customers and facilitates customization.

Entity in CRM – An entity is like a table that hold certain type of data with attributes.

Ex: Lead, contact, account, case, opportunity etc.

Attributes of the Contact Entity will be – Contact name, Guid, addresses, owner, Type of contact etc.

## Different Entities in CRM

Lead – A potential individual (unqualified) who searches in public site for insurance products.

Contact – User makes entry of this person (interested in taking insurance policy) in CRM.

Types of contacts: -

* Suspect – The person may or may not choose to be policy holder.
* Qualified suspect – The person is aware of insurance product
* Prospect – Represents a company that has real potential to be new Client.
* Client - Who purchased the insurance product and became the confirmed customer.

Household – Grouping of two or more contacts (family/group) with 1 contact as primary and remaining as dependents (child, spouse etc.).

Business/account – A company taking insurance for a company. Each business will have one contact (mandatorily) as primary.

Quick campaign – An advertisement

## CMR terminologies

LE – Unique identifier for any record and used in Enterprise. Like policy Id. Ideally all contacts should have LEId. Need to check if not generated.

Life Cycle Code = 1 (for prospect), 2- for client (HO record= Yes, ideally)

LeSrcCode – LE type.

BaseLE- Unique identifier for a contact in Enterprise. BaseLE should be generated else need to check why it’s not generating.

Owner – Always a team and defines role.

NetworkClientInd – Used to find out if user belongs to NO (true) user else normal user.

NOTerrNum – Used to identify the NO user in CRM.

DsbNum (Like agent #) – For non-NO user, we need to enter this when NetworkClientInd = false

OperatorIdNam (LAN Id) - For non-NO user, we need to enter this when NetworkClientInd = false.Different addresses: Home, business, NM address, More address

for Business - AddressLine1 will be NM address

For contact – AddressLine3 will be NM address

Each member in household can have different business unit.

Owner – A group/team to which user (creator of contact) belongs to. Ex: CRM CORE 010101

Administrator – Each group/team will have an administrator

Team category – FR, AFR, TechCoach, CRC, Supervision, Leadership, CFS

How to create team:-

* CRM online🡪 Security🡪 Teams🡪 NEW
* Team name= **Naik, Ashok Pre-contract**, Administrator= Nai7033, Team category= **Pre Contract**, Team type= Owner, Business unit= nmtestmigration2
* Manage roles: Set Field sales role
* For team category: FR, set Roles= Data import, HO Analyst

Business units: - Grouping of similar type if users to set different security roles for different users.

Users/ team belongs to some business unit.

Pilot user: users store in the database and these users only can launch PPA online.

Non pilot users will not be stored in the db. So for that to launch we need to have XPPA installed.

Types of BU:

Parent unit is always **NM**

NO – NW office (direct child of NM)

DNO - District n/w office (can be direct child of NM or child of NO)

CUD – College unit (child of NO)

Ex of NO: Dino’s Network office, Mack network office

To Check for Owner role in CRM

* CRM- Settings- Security
* Teams – choose any team and right click – Open in New window
* Change Guid to required one (owner guid got from log file or CIS debug)
* Manage role - Check for role – If not exists- add role (examples below)

Role = System user (for normal user)

Sys admin (for admin access)

Field sales – For FR (admin of the team, AFRs are members of FR)

# CIS (Contact Integration Service)

Contact Integration service is a Web API for lead management. It has the 2 main functionalities – Create contact and Get contact.

Create contact: It creates a contact in CRM based on the contact request (sent in JSON request).

Request validation will be done for fields like FirstName, LastName and LegalEntityId

Get contact Guid by calling CreateQualifiedSuspectContact(CRM entity, owner)

Add suspect Id and Adobe Id in ContactIntegrationMapping.

Return Json response with contactGuid and ContactOrgUrl.

Get contact: Fetches the contact record from CRM based on – BaseLE and GUID

GetContactByBaseLE

GetContactByGuid

Some more details-

Namespace Microsoft.Xrm.Sdk.Client – Is a way to access CRM in test project.

Create proxy for OrganizationServiceProxy (of client CRM) and access Retrieve ()

Pass Entity name, Guid and ColumnSet to retrieve all the fields of contact record based Guid.

Note: ColumnSet(true) = Select all the fields of the contact record.

# CER (CRMContactEventReceiver)

CER is a WCF service used by Enterprise (core business).

PreProcess 🡪 In PreProcess stage we check BaseLegalEntityId, LE ,LESourceCode and owner should exist in the request object. If not exists then "Mandatory params are missing" with response code=400.

Check if owner exists: -

Using NetworkClientInd we came to know owner Type . Check user is exists in the system. We get primary team from owner,owner type. If Owner type = FR then then we get team from owner LE and LESrc. If owner type = CRC then we get team from territory number.

Process() 🡪 This method is overridden and invokes the Add Contact operation

addContactFlow – Can be Add/Update for AddContact

if addContactFlow = false , then Update Contact

**Inbound–Contact**: Contact will be directly created in Enterprise (ex: using CER) and no need to wait for BaseLegalEntityId generation.

**OutBound-Contact**: Contact will be created using CRM UI and BaseLegalEntityId generation takes 2-3 min to generate.

WCF service launch errors/solution

1. Error - [Could not find a base address that matches scheme https for the endpoint with binding WebHttpBinding. Registered base address schemes are [http]](https://stackoverflow.com/questions/9817718/could-not-find-a-base-address-that-matches-scheme-https-for-the-endpoint-with-bi)

Resolution:- Change web.config of service project as below.

1. <security mode="None">
2. <serviceMetadata httpGetEnabled="false"/>
3. <endpoint address="mex" binding="mexHttp~~s~~Binding"/>

# FER (Field Entity record)

$/FSP-CRM/Branches/CRM\_dev/crmintegration/CRMFieldEventReceiver/CRMFieldEventReceiver

There are 3 stages in FER.

1. Pre-process: Common class to create instances of classes. (Factory pattern)
2. Process
3. Post-process

Services available in FER

1. AddBusinessUnit– Biz unit is a logical grouping/company.

Every Biz unit has a manager. (Not a mandatory field) in CRM.

If business unit status in I in request, then set BU as I – inactive

1. DeleteBusinessUnit– Le Id of biz unit, Le source
2. UpdateBusinessUnit
3. Add user- pass LE id,

JobFuntionCodes (passjob fun code, FldOrgLe)

Chk any role exists for the providec JobFunCode

By default, user will be disabled, use DisableEnableEntity

Set role for user.

1. DeleteUser
2. UpdateUser
3. AddTeamMember – Create team, Add user to team (teams are like – CRC, FR, TechCoach, Supervisor etc.)
4. RemoveTeamMember
5. UpdateUserHierarchy
6. RefreshUserRole
7. UpdateUserState

# Config Entries

1. (a) Add entry in CRM Configurations settings - **Do it only for Dev**

CRM 🡪 settings -> ConfigSettings

Active config settings

Add New 🡪 Key = EmanagerUserId, value=Zeta\_Internal, Description = EmanagerUser (for ex.)

(b) Edit: edit key/value as applicable.

\*Get GUID 🡪 Choose Email link 🡪 Copy guid (after %7b and before %7d)

\*\* GUID will be same for all the environments.

1. Add/edit config.xml (Non-prod and Prod) – Do it for all environments
2. Get ConfigData\_Dev7.zip from below Tfs path

**$\FSP-CRM\Branches\CRM\_dev\mscrm\mscrm\Builds\15.0.0.0\ConfigData**

**\***Please change build number to latest build/PI#.

1. Check out ConfigData\_Dev7.zip, copy to local path, unzip and edit data.xml for new/existing entry, Zip and replace .zip file and check-in.

Similar changes to be done for all the below configs.

1. ConfigData\_Dev7.zip
2. ConfigData\_Dev6\_VEGA.zip
3. ConfigData\_TEST7.zip
4. ConfigData\_TEST6\_VEGA.zip
5. ConfigData\_TRAIN.zip
6. ConfigData\_STAGE6.zip
7. ConfigData\_STAGE7.zip
8. ConfigData\_PROD.zip
9. ConfigData\_CRMTNMCPERF.zip
10. ConfigData\_CRMTNMCINT.zip
11. ConfigData\_CRMTNMC.zip
12. ConfigData\_TEST6POC

Note:-

Record\_id = <guid>

Value = <guid>

<Nm\_key value= Emanager\_Id>

<Nm\_value value= Zeta\_Internal>

\*nm\_description: Normally taken same as nm\_value.

# BAM (Batch Attachments Manager)

BAM is a windows service, scheduled to run @11PM daily.

On TFS- $\FSP-CRM\Branches\CRM\_dev\crmbam\BatchAttachmentManager\BatchAttachmentManager\CRMBAM.sln

## Functionality: -

* Fetches attachments from CRM (using attachment Guid)
* Converts CRM entity into attachment Entity
* Passes attachment Guid to publisher where publisher makes entry of attachments data into DB.
* Delete Attachment from CRM with attachment Guid.

## Testing BAM:-

* Close VS
* Run Visual Studio\_bat.bat (as Admin), enter password
* Open BAM Project by proving Prod credentials.
* Run test cases (enable Resharper 🡪 Rt.click on NM.Attachment.NUnitTest🡪 Debug unit tests)

## BAM deploy: -

Download crmbam.zip from Artifacts in TeamCity\Projects (after check-in)

Put crmbam.zip in **NTAPTH7837M00\BAM (For CRM Dev)**

Unzip and run setup as Admin

Press 🡪 N

setup.bat - runas admin

release folder gets all the file

compare bam attachment

BAM is a windows service-> run-services (will get BAM in the list)

TFS WorItems (Sprint link) - http://tfsprod.nml.com:8080/tfs/NMCollection/FSP-SAFe/CRM-Usual%20Suspects/\_backlogs/TaskBoard/PI%2013/Sprint%201%20-%20PI%2013?\_a=requirements

Press any key to exit.

Verify the build status in TC 🡪 Agents

# E-Manager

E-manager (Email blast App): -

$/FSP-CRM/Branches/CRM\_dev/mscrm/mscrm/Source/NM.Crm/Workflows

This App is used to subscribe campaigns to contacts and send Emails.

User Email is mandatory for E-manager

## E-Manager App functionality

* Choose contact(s) in CRM🡪Choose E-manager subscriber from more options
* Select owner and 1 or more campaigns
* Choose subscribe
* If no errors (like in blow example) then selected contact(s) will be subscribed and Email is sent.

\*Contact must have Email ID as contact should get intimation after subscribing.

\*Contact must have Email ID as contact should get intimation after subscribing.

\*Example of subscribing contact (with error messages in Red).

|  |  |  |  |
| --- | --- | --- | --- |
| A Jenn |  | no email | |
| a | ch | devikuppili@northwesternmutual.com | belongs to another FR | |

## Web resources

CRM-> Settings-> Customization🡪 Customize the system--. Entities, Web resources etc.

Web resources can contain .js, .png, .html, Xml etc.

We can add new resource file (New button)

Or we can edit the content of files (Choose resource-> text editor-> Save)

All the added/ modified resources need to be published (Publish button)  
Web resources from Dev environment can be exported to test environment using Export solution option.

Q: How to create solution?

We create solution in CRM whenever CRM customization is required. We need to keep CRM Web resources (.js, xml, Html, img updated w.r.to CRM\_Web\_Resources.sln)

CRM – Settings – Solutions – New –

Display Name = CRMTestSolutuion

Publisher = <Choose Default publisher>

Version = 1.0.0

Add the existing items – Web resources (.js, htm, png, .css) ex: NGISRequest.js, Library\_HHNGISRibbonLaunchInContext.js

\*To edit the files for new code changes, double click the file (ex: NGISRequest.js) – Text Editor – Copy content from NGISRequest.js (from CRM\_Web\_Resources.sln).

Add Entities – Ex: Household, contact etc. for which say Launch NGIS functionality is required. Export solution after adding entities

Publish All customizations

# Datapower

IBM Datapower Gateway is a single multi-channel gateway that helps provide security, control, integration and optimized access to a full range of mobile, web, application programming interface (API), service-oriented architecture (SOA), B2B and cloud workloads.

Set up Rapid SQL:-

Datasource🡪 Register Datasource

DataType = IBMDB2

Register Type = Use alias from IBM client

Server = CRMTRNT

DS name = CRMTRNT

Security params

UserId = Unix Id (LanID) and Pwd

# AD Explorer

Open AD explorer.exe

Search🡪 Class= common classes🡪 Attribute = Cn 🡪 value = nai7033🡪 Add 🡪 Search🡪 Double click search result

CN=NAI7033,OU=DIV26,OU=DPT13,OU=HO,OU=Internal,OU=Personal Accounts,DC=nm,DC=nmfco,DC=com,<default domain> [ntdcph9823m00.nm.nmfco.com]

Double click Member of 🡪 Copy the list of AD groups provided with access.

# CRM Plug-Ins

TFS Path- $/FSP-CRM/Branches/CRM\_dev/mscrm/mscrm/Source/NM.Crm/Plugins - RestructuredSolution.sln

* Plug-ins are triggered automatically for the operations (ex: Create, update, delete, share contact etc.) performed in CRM on the entities.
* Plug-in is an out-bound system as it pushes data into Enterprise.
* Plug-ins are linked to Enterprise via Publisher (PublisherServiceBaseWrapper)
* Execute method of IPlugin will be implemented to perform the actions.

Plug-in are used-

* To fetch data from CRM and Add to Enterprise.
* Get the Source URL. Etc.
* To Call publisher

There are various plug-ins like below -

NM.Crm.Plugins.CreateOrUpdateContact 🡪 AddContactToEnterprise, PreAddOutlookSynchCheck, PreUpdateOutlookSynchCheck, UpdateContactToEnterprise

Where, ‘CreateOrUpdateContact’ 🡪 Whenever new contact is created or the contact is updated, this plug-in will be triggered and same operation is performed in Enterprise.

NM.Crm.Plugins.SystemUser

NM.Crm.Plugins.Phonecall

NM.Crm.Plugins.Merging

NM.Crm.Plugins.PhonecallReactivation

NM.Crm.Plugins.PhonecallLogic

\*Legal Entity , contact , business etc. are available at -

~\CRM\_dev\mscrm\mscrm\Source\NM.Crm\Plugins\NM.Crm.Plugins.Common.JointWork\Helpers.

# Plug-In Registration tool

|  |
| --- |
| SDK🡪Bin🡪pluginregistration.exe |
| Discovery Url=https://crm7dnm.nml.com (For Dev) |
| Connect using dev admin-> Choose CRM7DNM🡪 Double click open plug-in/workflow |

This tool Is used to update/register/unregister DLLs, Debug plug-in, search plug-in.

Configure Plug-in step includes below parameters

* Message (the action to be triggered)
* Primary entity - The entity (Lead, Contact, account) that must be processed by plug-in.
* Event handler (Name of the plug-in)
* Execution workflow for plug-ins will be set in Plugin tool (based on the required Execution order ex: 1,2,3)
* Stage of execution – Pre-validation, pre-operation and post-operation
* Options available for Execution mode – Asynchronous (for post operations), Synchronous (execute immediately, used for pre-operations).
* Deployment mode – Server, offline

\* InputParameter values – Depends on the Message and Stage used to register the plug-in.

\*Output parameters only exists in Post stage.

Ex: - context.InputParameters["Target"] is present in I/P parameter for Create/update messages.

* Delete request has Target property and its type is EntityReference.

Options available related to registration of assemblies are -

* Update : For existing DLLs
* Register : For New DLLs

Procedure to update workflow:-

* Choose WF/assembly name for update (for ex: NM.Crm.Workflows.LeadManagement)

Process name = NM.Crm.Workflows.LeadManagement.QualifyLead

* Right click WF🡪 Update
* Choose assembly to load (C:\Ashok\CRM\_dev\mscrm\mscrm\Source\NM.Crm\Workflows\NM.Crm.Workflows.LeadManagement\bin\Debug)
* Click update selected plug-ins

Below target location of DLL can be chosen

* Database
* GAC
* Disc

# Code flow for Merge Plug-in Functionality

NM.Crm.Plugins.MergeEntity - Involves 2 plug-ins

1. FetchLegalEntiesForMerge

Execute () implemented from IPlugin will have below logic.

* Obtain execution context from service provider
* Obtain source and destination GUIDs.
* Use GetLegalEntityIdsForGuid (of PublisherServiceBaseWrapper) and get source and Target LEs into EntityCollection
* Set the attribute for source and target entities

sourceEntity.Attributes[EntityAttributes.MERGE\_LEGALENTITYLIST] = sourceLE;

targetEntity.Attributes[EntityAttributes.MERGE\_LEGALENTITYLIST] = targetLE;

1. MergeLegalEntities

NM.Crm.Plugins.Merging – Involves 3 plug-ins

1. DeleteSubordinateRecord
2. PreventMergingDifferentOwners
3. PreventSubordinateHORecordMerge

## Assign, share and unshare contacts on CRM

### Assign contact

* Create a New contact using WebApi/CIS (using postman), with owner =

[SULLIVAN,J 1234](https://crm7dnm.nml.com/_root/homepage.aspx?etc=2&pagemode=iframe&sitemappath=ClientManagement%7cWorkplace%7cnav_conts" \o "Open SULLIVAN,J 1234" \t "_self) (for example)

* Choose this contact and select ‘Assign’ from more commands (in CRM UI).
* Assign to = User or team
* User/team = KIRK, DONALD 051685 (for example)
* When we login CRM with kirk account, we will be able to see this new contact (as well as in Admin login (crmadmdt).
* After assigning the contact to different user group, only 1 contact exists in CRM (with ownership changed to new owner).

### Share contact

Sharing contact (SULLIVAN,J) with KIRK, DONALD

* Choose the contact🡪 select Share from more commands🡪Look for=Team🡪 Search= Kirk🡪 Select 🡪 Add
* Choose permissions for Kirk (By default Read access is selected) 🡪 SHARE
* Now this contact can be seen in Kirk’s login also. To cross check the share🡪 click on first name🡪 We can see Joint Work: KIRK, DONALD 051685 (But owner will be SULLIVAN,J)
* Joint work info will be visible to the owner only with whom the contact is shared (but not visible to Admin).
* Contact will be shared with different owner group and contact can be seen with Original user SULLIVAN,J
* New owner (with whom contact is shared) will be able to read/write/share/assign (based on the permissions selected during share)
* Contact can’t be shared if owner belongs to non-integrated team.

### Unshare contact

* Unshare can be done in Admin the login
* After unshare ownership of contact will be reverted to SULLIVAN,J (from Kirk Donald) so Kirk will not be able to view/write/share/assign.

### Merge contact

* Merge operation can be performed only when the contacts have same owner/OwnerID.
* If we select Target/Master and Source/Subordinate contact, after merge master record will be retained and source/subordinate contact will be deleted.
* In case, mobile# for Master is empty and source has some mobile#, during merge if we choose mobile# (by choosing mobile# radio button from Source), after merge mobile# present in source will be copied to Master. Similarly, other fields (if chosen).
* All types of contacts can’t be merged. For ex: Client type (HO record) can’t be merged with other contacts. If source is HO record, then we can’t delete it.

# CDAS – CRMData Access Service

Source control 🡪 $/FSP-CRM/Branches/CRM\_dev/crmdas/CRMDataAccessService

It’s a WFC service which is used to get Household (HH, Family or a group) details from CRM. Various users can get HH details from CRM.

HH data can be retrieved in below ways-

* HH LE
* Contact LE
* HH Name

User can be identified using -

* LE Id
* Distribution Number (in case of NO user and when NetworkClientInd = true)
* Operator Id (like LAN Id) used when NetworkClientInd = false (non-NO user)

\* As per prospect journey, HH service is going to be a WEB API instead of WCF service (CDAS).

\* Refer to SOAD UI request xml for more info.

\* One SOAP response can have multiple HHs.

\* One HH can have 1 or more contacts

# CDAS- Code flow

* Execute GetHouseholds() of CRMDataAccessService.svc.cs
* GetHouseholds() calls GetProcess(EntityType.Household)).Execute(request)
* GetProcess() of Process factory based on the EntityType passed to GetProcess (Entity type= household in this case).
* Post this Execute() is called. Execute takes request (GetHouseholdsRequest)
* An instance of CRMBusinessUnitServices is created in HouseHoldProcessFlow
* CRMBusinessUnitServices \_businessUnitServices = new CRMBusinessUnitServices()
* CRMBusinessUnitServices creates instance of CrmUtils and this calls constructor CRMInstance.

CRMUtils \_crmUtils = new CRMUtils()

* In CRMInstance , Service proxy is checked whether it already exists in CRMcache
* If proxy does not exists in cache then SetupServiceProxy() is called.
* CRM proxy of Dev/Test is created based on Key=OrgServiceUri under the section <CRM> in the constructor CRMInstance(). Below ClientCredentials are used

|  |  |
| --- | --- |
| CRM7Dev | <https://crm7dev.nml.com/CRM7DNM/XRMServices/2011/Organization.svc> |
| CRM7test | https://crm7tnm.nml.com/CRM7tNM/XRMServices/2011/Organization.svc |

ClientCredentials.UserName.UserName = "NMTEST\\crmadmdt"

ClientCredentials.UserName.Password = <test\_password>;

serviceProxy = new OrganizationServiceProxy(uri, null, cred.ClientCredentials, null);

* After creating the proxy, GetUserDetails() of Process() is called where user details (Guid) are fetched by passing the request.

<icr:UserIdentifierInfo>

<!--Optional:-->

<icr:LegalEntityId/>

<!--<icr:DsbNum>045545</icr:DsbNum>-->

<!--Optional:-->

<icr:OperatorIdNam>nmtest\pure6658</icr:OperatorIdNam>

<icr:NetworkClientInd>false</icr:NetworkClientInd>

<!--Optional:-->

<icr:SenderApplicationNam/>

</icr:UserIdentifierInfo>

* If Guid is Empty or If NetworkClientInd = false , it’s the case "No User Found"
* RetrieveHouseHold(guid), CRMHouseholdService. GetHouseholdForLegalEntity() gets all the Guids of HH.
* Another option is to get HH by GetHouseholdForName(request1.HouseholdName,…)
* GetHouseholdDetails(request, guid, HH), returns household.Contacts
* If user has no access then AccessViolationException
* When response from RetrieveHouseHold() is empty then "No matching Households found".

# Launching various Apps from CRM

TFS path - $/FSP-CRM/Branches/CRM\_dev/mscrm/mscrm/Source/NM.Crm/WebResources

* We have GO TO option in CRM to launch/navigate to various Apps.
* We can launch various Apps (SRC, PPA, CSS, E-files, NM connect, LinkedIn, SRC work que).
* On launch of Apps, the required data from CRM will be fetched and populated for these respective Apps.

Brief about these Apps

PPA:- Personal Planning Analysis

* Can be launched from CRM
* PPA has an option to customize
* PPA issued only for Households. PPA will not be applicable for contacts.
* PPA client will be household member

CSS - Client service system

SRC – Service request center

Linked-in – Applicable for contacts but not for households

# PPA (Personal planning Analysis) Launch

PPA Launch has 2 options – Launch PPA online and offline.

Online launch – PPA can be launched for the existing clients by appending security token to PPA Url. Pilot user has the access for online launch (

All users will not have access to launch online access.

clientId = 2496680 (Owner dsb number)

PPASecureLaunch URL (CRM-Config setting) = <http://ws59stage.nml.com/fielddplan/securelaunch>

PPA Enabled = Y (Enables PPA Launch from House Hold)

Offline launch – Non-pilot user can launch PPA for HH. We need PPA launch installable on the machine otherwise PPA launch just downloads the file instead of launching HH in PPA windows app (for offline launch).

PPA client page has 2 options for launch

Existing PPA client

Create PPA from HH

How PPA launch works:-

* Library\_HHRibbonLaunchInContext.js

ppaClientURL = "/WebResources/nm\_/html/SearchOrCreatePPAClient.html"

On GoTo PPA , gotPPAConfig() gets all the required data and PPA window will be opened.

window.open(ppaClientURL + "?Data=" + parametersURL, "", "width=1000, height=680, left=200, top=50, menubar=no");

~\WebResources\htmlClientPPA.html (in CRM customization file name is set as – SearchOrCreatePPAClient.html)

//Goto PPA button click event handler

$('#btnGoToPPA').click(function () {

$('#notification').html('');

setTimeout(function () {

launchPPARecord("PPAExisting\_");

});

});

* Choose HH member (make configuration changes based in need)
* GoTo PPA.
* In case of offline launch – xppa file (Ex: CRMPPA92374.xppa) will be created which will have details like client Id etc.
* This client Id can be used to launch PPA online by appending client Id with PPA Url.

Please note:-

* Atleast 1 member must be a client.
* Only 1 HH member can be made as PPA client
* State is mandatory for Client.

# CRM Proxy

TFS Path - $/FSP-CRM/Branches/CRM\_dev/crmintegration/CRMProxy

CMR proxy is a RestFull WCF service

This is used when calling services from javascript.

Use CRM proxy Url in PPALauncher.cs (for PPA launch)

(~\CRMProxy\NM.Integration.CRMServices.CRMContactValidationProxy\Process\ PPALauncher.cs)

# Urban code Deploy

It’s used to deploy Projects in other environments (In 7Dev – auto deploy happens after TFS build succeeds with 100% test case pass).

Procedure to deploy

1. Go to Applications tab
2. Select CRM
3. Choose the environment to deploy – ex: Dev02 (for 6Dev), Test (7Test)
4. Press Play button (response process)

Process = Deploy

Choose version

1. Select Project (ex: CRMHouseholdIntegrationService)
2. Version to Deploy🡪 Add🡪 Choose ‘Latest available’ from ‘Version lookups’ (Uncheck this)
3. Submit deployment

**PUTTY & Rapid SQL**

Check in which all unix groups we have access -using Putty

Keyboard – Ctrl-H

Hostname = bastila

Choose bastila and load

Login as – LanID (unix Id), pwd

Type who am I

*nai7033 pts/69 Apr 25 04:16*

*(/home/nai7033) 7>* groups

//I am in the below unix groups.

*cmlib3cs crmbam crmcel crmcvs crmetl crmfel crmhrt crmpub crmtrnap crmtrnas crmtrnat crmtrnsp crmtrnss crmtrnst crmutil users*

//To check unix groups for other user: example

*(/home/nai7033) 7>* groups Aga7673

Setup Rapid SQL:-

DB2 UDB Instance Information:-

<http://ws1.nml.com/udb/udbInstData.html>

To setup DB2 datasources- we need to install DB2

Install DB2: Version 10.1 Fixpak 3\_Runtime Client 64

<http://mutualnet.nml.com/is/udb/cae.htm>

Update catalog - W:\appls\NMLSOFT\UDBCAT8\08.00.00.01\ INSTNM02 – Rt.click and run as admin

*\*Close Rapid Sql before updating catalog*

Setup data source in Rapid Sql

Rt.click on DB2 servers 🡪 Register (Or Datasource-register Data source)🡪Database Type = IBM DB2

Connection info- registration type = Use alias from IBM client

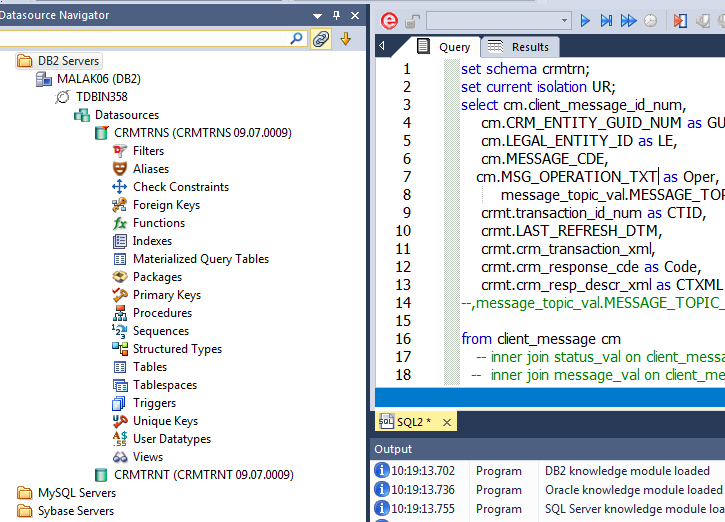
Server = CRMTRNT or CRMTRNS etc. will be auto populated if catalog is updated

Ex: Server = CRMTRNT (remote node=Malak06, port=09.07.0009)

Security parameters: UnitId (NM Id)/prod pwd 🡺 test connection – should succeed

🡺Finish

After registering DB2 of rapid SQL should look like…



# SOAP UI setup

For CDAS project:-

Open SoapUI -> File-> Import project -> Choose NM Basic Template.xml

(Get exposed URL from WSRR details excel-> Add WSDL🡪<https://nmsbtest.nml.com:8443/CRMDataAccessService/CRMDataAccessService.svc?wsdl> )

Rename project as CDAS-WSRR. WSRR stands for Web service registry and repository.

Double click project -> In properties

ConsumerId =CRMADMIN\_02 (For Dev)

ConsumerId\_SSL =CRMADMIN\_02

In WS-Security configurations ->Choose Name – UNT-> UserName and provide NmTest Id/pwd

Service namespace = <Get it from sheet [Services] of [**WSRR Details.xlsx**](http://project.nml.com/sites/TNC_Renewal_Prgm/program/Shared%20Documents/WSRR_Details.xlsx)**​ for CDAS**> (<http://nml.com/services/information/ICRMDataAccessService>)

ServiceEndpoint = https://nmsbtest.nml.com:8443/CRMDataAccessService/CRMDataAccessService

Exposed URL = https://nmsbtest.nml.com:8443/CRMDataAccessService/CRMDataAccessService

ConsumerIdUsed = ${#Project#ConsumerId}

NamespaceUsed = ${#Project#ServiceNamespace}

Rt.click WSRR Lookup and Call-> Add step-> Soap request

Check all the check boxes

Rename test request with GetHouseholds (for example)

Add proper header and request body-> Validate the request

|  |
| --- |
| <soapenv:Header> |
| <wsa:EndpointReference xmlns:wsa="http://www.w3.org/2005/08/addressing"> |
| <wsa:Address>${#Project#ServiceEndpoint}</wsa:Address> |
| <wsa:ReferenceParameters> |
| <gep63:consumerIdentifier xmlns:gep63="http://www.ibm.com/xmlns/prod/serviceregistry/profile/v6r3/GovernanceEnablementModel">${#Project#ConsumerIdUsed}</gep63:consumerIdentifier> |
| <gep63:contextIdentifier xmlns:gep63="http://www.ibm.com/xmlns/prod/serviceregistry/profile/v6r3/GovernanceEnablementModel">${#Project#NamespaceUsed}</gep63:contextIdentifier> |
| </wsa:ReferenceParameters> |
| </wsa:EndpointReference> |
| </soapenv:Header> |

Point the request to proper Url- Choose Edit current from dropdown-Edit Endpoint => ${#Project#ServiceEndpoint}

Double click on GetHouseholds request 🡪 Choose **Auth** (bottom left) 🡪 Authorization= Basic🡪Outgoing WSS = UNT (which is set under WS-Security)

Double click WSRR Lookup and Call-> Run this test case

Choose Step3 [getHouseholds]-> Double click-> Verify Response Message 9for any error or success case)

# CRM Solution (Release) changes

Create a new solution or use an existing solution (ex: Temp) and add the required JS files or entities (contact, household etc.)

(also Get the **backup** of Solution file (ex:Release 15\_1)-> export solution(.zip) or take screenshot)

~~CRM->Settings-> Solutions-> Choose Release 15\_1 (or as applicable)->Component-> WebResources->~~

~~Export solution-> Next->Next~~

~~We get a Zip file like Release\_15\_1\_15\_1\_0\_0.zip~~

CRM->Settings-> Solutions->Search for \*temp solution->

Entities: Add the Entities for which customization has to be applied. (ex: Entity->Add entity->Add existing->contact, Household

Web resources: Add existing-> Add GenerateQueryString.js (for ex)-> double click -> open in text editor-> Copy the whole .js file (of CRM\_Web\_Resources.sln) for which the required changes are done (as part of the requirement).

Export solution-> Next->Next

We get a Zip file like Temp\_5\_2\_3\_2.zip

Unzip and edit customizations.xml as per requirement (ex: renaming of GoTo button)

Note: Rename <Title description=?> for NMConnectButton.LabelText, linkedInButton.ToolTipTitle, NMConnectButton.Alt etc.

Zip it back with the updated customizations.xml

Import solution-> choose the Zipped file-> Import

\*Add the same Entities, web resource (.js) files to the actual/latest solution (ex: Release\_15\_1)- Save and close.

Try if the changes work (ex: check if Go-to button is renamed from ‘SRC work queue’ to ‘SRC Financial planning’)

NM.Tools:-

$/FSP-CRM/Branches/CRM\_dev/mscrm/mscrm/Source/NM.Tools

1. BaseLEandLEGenerator/ GenerateLegalEntity: Used for republishing the prospects (mainly by NMC team).
2. CRMContactShareTool (recent one): used for data fix in Prod so that share contact is updated in backend (like we can see joint work info in CRM).
3. AddTeamRoles

**Extract contacts with Guid details**

* Advanced fine->contacts-> [New]-> BaseLegalEntityId-> Does not contain data-> Results
* Click on Export contacts (top right corner)->Open excel-> unhide all columns-> Will find Guids in 1st column(Column:A).

WID details file:-

$/FSP-CRM/Branches/CRM\_dev/mscrm/CICD/test/e2e/Nightwatch-legacy-forms/data

Code freeze and PROD fix TFS branches:-

* Code freeze (happens 1 week before Prod date): $/Group84/Trunk
* Prod fix (Prod bug fixes): $/FSP-CRM/Branches/CRM\_pfix

Business space and SER Admin Tool

Pre-requisite:-

1) Verify WSRR Diagram has been created/updated - done

<http://project.nml.com/sites/TNC_Renewal_Prgm/program/Shared%20Documents/CRM%20WSRR%20Diagram%20-%20'Proxy'%20calling%20'PPAPlanClientService%20V4'.vsd>

2) Verify WSRR Details spreadsheet has been updated - done

<http://project.nml.com/sites/TNC_Renewal_Prgm/program/Shared%20Documents/WSRR_Details.xlsx>

3) Verify Consumer Contract is updated - done

<http://project.nml.com/sites/TNC_Renewal_Prgm/program/Shared%20Documents/Proxy_PPA_Plan_andClient_Service_Consumer_Contract.docx>

PPA (Personal Planning Analysis Plan and Client Service)

* Latest version of PPA is V4
* For any changes in PPA service (new methods, new parameter/param names, field changes etc.), configure/update service -reference under the project-CRMProxy.DataAccess
* Update web.config for all the enronments (Web.config, Web.Dev02.config, Web.Prod.config)
* Fix build errors (if any) for CRMPRoxy and CRMProxy.DataAccess projects.
* Check-in to deploy in Dev

Try to launch PPA from Crm\_dev

* Select HH🡪 Launch PPA🡪 choose existing client/create client🡪Go To PPA🡪 (<http://ws59stage.nml.com/fldplan/app>) will be launched for test/stage🡪 New Engagement🡪Next🡪Next🡪Plan Creation-Tax method=Simplified text🡪Next🡪 Choose plan model🡪Next🡪 Finish🡪 Plan ID will be created.
* Check Client Information for more details (name, address etc.).

Business space:-

Used to verify if WSRR setup exists for the particular service.

https://ws53.nml.com/mum/enabler#pid%3DUSER\_NAVTREE\_ROOT\_OID000000000000000%26

NmId/pwd

GoTo Spaces🡪 Search space with service name for PPA v4 service(ex: Personal Planning Analysis Plan and Client Service)

SER (Service Endpoint Resolver): -

http://sertest.nml.com/ServiceEndpointResolver/admin/index.jsp

No login required

Provide below details: -

(Sample values for PPA)

Consumer Id = CRMPRX\_01

Svc Namespace = http://nml.com/services/information/field/ppa/PPAPlanAndClientServiceV4

Choose Exposed and check if returns the exposed test/stage/prod Urls and check if the details match with the one provided by PPA team.

Result 🡪

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?><port xmlns="http://schemas.xmlsoap.org/wsdl/" binding="tns:PPAPlanAndClientServiceV4Binding" name="PPAPlanAndClientServiceV4"> |
| <soap:address xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" location="https://nmsbtest.nml.com:8443/fldplanService/PPAPlanAndClientServiceV4"/> |
| </port> |

**Transaction DB**

CRMTRNS DB is for 7test

CRMTRNT DB is for Dev

**How to test if service is working with ApiKey**

Ex: <https://api-int.nmlv.nml.com/v1/ngis-ms-suit/contexts/>

ApiToken: -

keyName= apiKey, Value= T64YGCvKhLRFhi79AfbPnDiyd20cQSCN

Note:- If apiKey is not supplied the we get error: {"code":401,"msg":"apikey not found in request"}

Result🡺

For SuitUrl = <https://api-int.nmlv.nml.com/v1/ngis-ms-suit/contexts/>

|  |
| --- |
|  |

GitLab

To add any file to existing project (ex: To add jenkinsfile to project)

Clone/download project to 🡪 cd C:\nmprojects

git clone <https://git.nmlv.nml.com/crm/crm-integration/CRMContactEventReceiver>

(To clone under crm-integration use:- git clone https://git.nmlv.nml.com/crm/crm-integration/BatchAttachmentManager src/crm-integration/BatchAttachmentManager)

(“src” root folder name is not mandatory to mention in the command)

create jenkinsfile and add it to C:/nmprojects/ CRMContactEventReceiver

Create branch (A new branch wil be created for each merge request)🡪 Publish🡪 Stage jenkinsfile🡪Commit stage🡪Sync🡪 Push🡪Create merge request

🡪Check remove source branch🡪Go to master

Git commit&push, Sync, Merge changes

* Do the change in project/solution
* Choose local branch ex: dev-ashok (by default it will be master)
* Rt.click solution - Commit- enter comment- commit&push
* Sync
* Push(Successfully pushed branch dev-ashok to origin.)
* Verify on GitLab UI (Repository-Branches) whether the new commit comment can be seen with latest date/time under **local branch**.
* Go to Project Repository- Files
* Create merge request
* Commit changes (cross if we can see the comment which was given during commit from Visual studio).
* Merge

Install NuGet packages for projects

* Choose the project and run below command (in Tools->NuGet package manager->Package manager console)
* **Install-Package Microsoft.CrmSdk.Extensions** (installs latest available version)

Clone from TFS

git-tf clonehttp://tfsprod.nml.com:8080/tfs/NMCollection/ '$\FSP-CRM\Branches\CRM\_dev\components\ExternalAssemblies' –deep

Jenkins:-

Copying configs and artifacts when MSBUILD12 is used.

stage ('copying configs and artifacts') {

bat 'md artifacts\\bin'

bat 'COPY "CRMFieldEventReceiver\\\*.\*" "artifacts"'

bat 'COPY "CRMFieldEventReceiver\\bin\\\*.dll" "artifacts\\bin"'

}

\*Where CRMFieldEventReceiver is Namespace of the projects from where config files needs to be copied to artifacts folder.

CRM Online

Online link

<https://ashokcrminfosys.crm8.dynamics.com/main.aspx#682077295>

Your user ID

[ashokcrm@ashokcrminfosys.onmicrosoft.com](mailto:ashokcrm@ashokcrminfosys.onmicrosoft.com)

pwd: me@111

From Developer discussion (31/Oct): -

<https://docs.microsoft.com/en-us/dynamics365/customer-engagement/developer/walkthrough-registering-configuring-simplespa-application-adal-js>

https://github.com/eisenbraun/columns

**Create/ Add user**

There are 3-ways to add/create user. Once user is added he/she will be able to login to CRM in the respective environment.

1. CF Portal (Consolidated Field Tester):

http://community.nml.com/sites/ITPS/gFHas/dmsap/cf/SitePages/CF%20Test%20Tool.aspx

1. CRM UI – CRM- settings-Security- Users - New
2. FER – user details- Add user (In SoapUI setup)

Input/request to add user: please add this user in crm7dnm wins1612/26194LWG

Here, wins1612 is FID and 26194LWG is WID (used in case of enterprise user)

**Approach 1:** CF Portal (Consolidated Field Tester):

(a)CF\_Portal\_Link <http://community.nml.com/sites/ITPS/gFHas/dmsap/cf/SitePages/CF%20Test%20Tool.aspx>

\*Make sure to choose Tools-> Enterprise Mode (In browser)

We will get the Xml template/Xml input as mentioned below

Xml input:-

By default <KeyType> will be LE Id, change it to WinsignIdNum, Agent number etc. (Refer [View Format Details](http://community.nml.com/sites/ITPS/gFHas/dmsap/cf/Lists/Inquiry%20Services%20Requests/Specific.aspx?FilterField1=Request%5Fx0020%5FMessage&FilterValue1=GetFieldRepPersonDetail001)])

Below is the replaced format with WinsignIdNum

<KeyType>WinsignIdNum</KeyType>

<KeyValue>26194LWG</KeyValue>

<NameTypeCde>4</NameTypeCde>

<AddressTypeCde>3</AddressTypeCde>

<EMailTypeCde>2</EMailTypeCde>

<IncludeOrgPersInd>N</IncludeOrgPersInd>

Take WID name (WinsignIdNum) and WID value

<KeyType>WinsignIdNum</KeyType>

<KeyValue>26194LWG</KeyValue>

(b) Click SEND (to verify user exist in DB🡪 We will get XML reply like mentioned below.

## Consolidated Field Tester - For MQ Services

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | **Env.** |  | **Action** |  | | **Queue** |  |  | putRequestGetReply = Synchronous request/replies  putRequest = Asynchronous put request on server's queue | | **Message** |  |  | getReply = Asynchronous and Subsequent replies  getNextReply = Get Subsequent replies for multiple packet replies | | **Msg ID** |  |  |  |  |  |  | | --- | --- | |  | **XML chosen (with input data populated) [**[**View Format Details**](http://community.nml.com/sites/ITPS/gFHas/dmsap/cf/Lists/Inquiry%20Services%20Requests/Specific.aspx?FilterField1=Request%5Fx0020%5FMessage&FilterValue1=GetFieldRepPersonDetail001)**]** | |  |  | |  | |  |  | | --- | --- | |  | **XML reply [**[**View Format Details**](http://community.nml.com/sites/ITPS/gFHas/dmsap/cf/Lists/Inquiry%20Service%20RequestReply%20Versions/AllItems.aspx?FilterField1=Request%5Fx0020%5FMessage&FilterValue1=GetFieldRepPersonDetail001)**] [**[**Select All**](javascript:document.getElementById('DebugText4').focus();document.getElementById('DebugText4').select())**]** | |  |  | | |

(c)Check Xml response to check if user exists.

If the user exists, you will be able to see the **XML reply** as in above screenshot**.**

(d)Verify user – To cross check

Open below Html file



* Choose Crm Transaction Type = AddUser
* Legal Entity = 20232348 (get it from above Xml response)
* User type = FR
* Submit Form

If the user exists, you will be able to see user details as mentioned below.

|  |
| --- |
| ON DEMAND REQUEST |
|  |
| Request Type: AddUser |
| Legal Entity: 20232348 |
| Person Type: FR |
|  |
| <?xml version="1.0" encoding="UTF-8" standalone="yes"?> |
| <AddUserRequest xmlns="http://nml.com/services/information/ICRMFieldEventReceiverServiceV1"> |
| **<LegalEntityId>20232348</LegalEntityId>** |
| <LegEntIdSrcCde>10</LegEntIdSrcCde> |
| <User> |
| <PersonFirstNam>Louis</PersonFirstNam> |
| <PersonMiddleNam>W</PersonMiddleNam> |
| <PersonLastNam>Garl</PersonLastNam> |
| <LineageTitleTxt></LineageTitleTxt> |
| <FullNam>Louis W Garl</FullNam> |
| <DistributorNum>026194</DistributorNum> |
| <OperatorIdNam>idea1035</OperatorIdNam> |
| <EmailNam>louis.garl@nm.com</EmailNam> |

(e)Add user: Perform this after performing the steps (a) to (d)

Use below Html file to Add user.



* Provide LE Id and set wait time to 90 sec.
* Click Submit Form

|  |
| --- |
| **Crm Transaction Type**   **Legal Entity**   **\*Related Legal Entity**   **Wait For Response?** No wait (submit and forget) 30 Seconds 60 Seconds 90 Seconds 120 Seconds  *\*Required when running "AddTeamMember" or "RemoveTeamMember" (LE = Assistant; Rel LE = FR)* |

(f) Assign team and set security role as system user for FR/AFR

- Choose user and choose any existing team else add new team

- Set security role as ‘System user’ for FR/AFR

\*For staff ID , choose team as CRC team

(g) Login to CRM UI and verify added user

- CRM-Settings-security-users- search for Nmtest\ <FID> = Nmtest\wins1612

- Check for fname, lname, Distributor Number, primary email, address etc.

1. Approach 2: CRM UI

Procedure to add/create user

1. Check if the given user exists in the mentioned environment (7dnm/7tnm etc.). If user does not exists then proceed with step2.
2. Get the details of the user (LEId, fname,lname,Business unit Id, Display Ind, Job function code(refer attachment below).

Fid = nmtest\wins1612

1. Add security role to user

Choose newly created user (dropdown)- security roles- Manage roles- Choose Fieldsales (if user is FR)

1. Create team

CRM- Settings-Security- Teams-new

Team name = <lastname, Firstname Dsb#> ex: ANKARATH, Deepak 773ANK

Choose team adim, team category=FR, team type=owner, business unit=NM

1. Set security role for Team

Choose newly created team (dropdown)- security roles- Manage roles- Choose Fieldsales (if user is FR)

Opportunity in CRM (Handled by CRM code team, opportunity is part of Sales)

The main purpose/goal of CRM is opportunity to buy/sell insurance products

How to create Opportunity

Client Management – Opportunity

Business/Contact = <Biz/contact name>

Opportunity status

* Inprocess
* Sold (customer is interested and bought the product)
* Lost (Customer not interested in buying the product)
* Onhold

Opportunity workflow

Case open/case preparation->Questioing -> Implementations-> Closed

Note:-

1. 1 opportunity for 1 Business/contact
2. Opportunity can only be created for Business/contact

Service: Opportunity

Service: case,

*Customer*: Contact, Account

Cacading examples:-

* Deletion of phone call is dependent on contact. If phone call is deleted then need to make sure attached contact is also deleted.

NuGet packages:-

To fix reference issues (if any) when we clone project from GitLab (an example)

* Go to references of NM.Crm.HouseholdIntegrationServiceTest
* Right click Manage NuGet packages
* Choose package and click on Restore or uninstall/install
* Make sure reference issues are resolved in test project after restoring Nuget packages

Fixing Test explorer issue – when project is not able to detect any test cases

* Go to references of NM.Crm.HouseholdIntegrationServiceTest
* Right click Manage NuGet packages
* Go to browse- search for Nunit 2.0 -> Choose NunitTestAdapter v2.1.1
* Go to package source settings button
* Under package sources- Click on + button to add new package
* Add “NM Nexus” package (refer below screenshot)
* Now, go to test case and try Run/Debug test case

|  |
| --- |
|  |

NMC launch issue (Prod)

Issue description: - The end user can launch to NMC from the Household in CRM but when they search for the household directly in NMC and try to launch it they receive a permissions error.

Here are the particulars :

User name: Cheryl Cranor

Household Name: Aaron Clutter Family

FR Team: Copeland, Shawn 078730

Steps to replicate – replicate the issue first in CRMTNMC.nml.com

* SSO login with full1189 (Hypke, George)
* Login to <https://crmtnmc.nml.com> with full1189 (Get the users for login from [crm-tests-users] from GitLab [https://git.nmlv.nml.com/crm/crm-platform/crm-tests-users/tree/master/data])
* Share contact with couple of FRs.
* Choose HH - Aaron Clutter Family and launch NMC connect (from GoTo)
* NMC App will be launched and search for above HH: We should be see HH (without any permission error). But as per issue currently there is permission issue in PROD (but not in tnmc)

Note: Activate the contact if status is Dead feed/inactive.

Contact : Yogesh, Jim Gallagher (CRM support team) for prod issues.

Azure portal, Service Bus

<https://portal.azure.com>

login as - [fname+lname@nmcop.com](mailto:fname+lname@nmcop.com)

pwd: nmpwd

In Azure portal

Resource groups

Queue/Topic (choose any one)

Subscriptions (broker/agent to hold messages)

Logic Apps: Gets messages from subscriptions, code will be in json format, if we want add new code we can use WebApi

Pushes message to MSMQ

MSMQ (instead of publisher)

Enterprise (fetches messages from MSMQ)

How to check for messages sent to SB

* Use ServiceBusExplorer.exe
* File- connect- SB Namespace- choose ‘Enter connection string’
* ConnectionString is = **Endpoint=sb://crmlab-integration-namespace.servicebus.windows.net/;SharedAccessKeyName=RootManageSharedAccessKey;SharedAccessKey=PBaWw+jb9uEDHqyfeU3iqLnoOuY7UPT1EwkIT3UvMIM=**

2-ways of calling SB

* By adding code in Plug-ins
* From plugin registration tool – by adding step

SB in Azure portal

Logic App

[Resource groups](https://portal.azure.com/)🡪[crm-lab](https://portal.azure.com/)🡪[crmlab-nmttestpocphase1-merge-contact-la - Logic App Designer](https://portal.azure.com/)

Topic name: crmlab-integration-topic

Subscription: crmlab-nmttestpocphase1-merge-contact-s (for Merge Entity- contact)

crmlab-nmttestpocphase1-assign-contact-s (for AssignLE-contact)

Existing technical design for MergeLegalEntity plug-in/publisher.



Create personal repository(fork) in GitLab

Go to any project-ex:HIS

Choose Fork

(A fork is a copy of a project.   
Forking a repository allows you to make changes without affecting the original project.)

* Click on Fork
* Click on name/picture
* Copy the changed git repository link
* Clone the repo
* Open repo
* Set personal path – No
* Create branch ashok\_online
* Pull from origin or dev\_online
* Push personal repo
* Submit merge request
* Target branch= dev\_online

Forked project looks like

|  |
| --- |
| [NAIK, ASHOK / HouseholdIntegrationService](https://git.nmlv.nml.com/NAI7033/HouseholdIntegrationService) |
| HouseholdIntegrationService |

* Delete existing fork

Choose Fork project- Settings – advanced settings- Expand (choose remove/rename/archive etc.)

Choose Remove project – type project name (ex: contactintegrationservice)- Confirm

Logic App

For MergeEntity plugin

1. Fetch legal entity- follows old plugin design
2. Merge entity – main merge – follows logic app
3. Data archival (Sql)- follows logic app (with Sql db)

Main merge flow

Scope2

Choose action- Http+swagger

Swagger Url = https://crmlab-

\*IsintegrationTeam

If{} else

Dynamics365

OrgName= nmttestpocphase1

Entity= Legalentity

Crm qry = $filter- contactid

Ex: contactid=<get from schema>

\*Get keyvault Url from security api-app code

\* Api apss can be published using – project – rt.click publish

\* restricting IPs🡪 App services- networking- IP restrictions

======================

Sql configuration for Data archival

Scope2

After if {} else

Add action

Add connector-Sql server

Connection = CRM-Integration-DataArchival

DB = nmttestpocphase1-integration

Connection user name = crmint1d

Pwd = <Add/call keyVault api-app to retrieve pwd> (can be hardcoded for time-being)

* Swagger Url = Crmlab security crmintegration apiappazurewebsites.net/swagger

Logic App

Merge Entity – contact , account

Different symbols/attributes of LA

* Scope, action, condition, filter array,…… etc.

Concept of user and team

User –

Inbound user – is the integration user and not allowed to send msg to publisher– ex:Crmint1d (client credential) as we can see in case of CER, FER .Entity coming from enterprise.

Outbound user- is the non-integration user and allowed to send msg to publisher

Ex: User of CRM UI - FR/AFR

Team- User belongs to some team

FER debugging

Hit this Url - <http://localhost:2430/CRMFieldEventReceiverService.svc>

For metadata related error do below changes in web.config

<basicHttpBinding>

<security mode="None"> (old= <security mode="Transport">)

binding="mexHttpBinding" (old: binding="mexHttpSBinding")

* Run FER in debug mode and hit the local Url. Expected result is success response code(200).
* \*All the refence projects/Dlls wil be in [crm](https://git.nmlv.nml.com/crm)->[[https://git.nmlv.nml.com/uploads/-/system/group/avatar/624/integrate.png](https://git.nmlv.nml.com/crm/crm-integration)crm-integration](https://git.nmlv.nml.com/crm/crm-integration)-> [components-CRMSharedAssemblies](https://git.nmlv.nml.com/crm/crm-integration/components-CRMSharedAssemblies)

**How to check if message is sent to Publisher from logic App**

* Go to LA- crmlab-nmttestpocphase1-merge-entity1-la
* Double click on Overview\Run history message
* Get"TargetBaseLegalEntityId": "12531184" OR  "TargetGUID": "a794ef44-7505-e611-80ce-00505686132b" (take only target details as source will be deleted) from HTTP step/ Input Body.
* Search in publisher logs for any message

Find in files- Find= a794ef44-7505-e611-80ce-00505686132b

Filer = crmpub\_debug\_\*.log

Directory = \\hotest\dfs02s\nmlstage\crmpub\logs

**How to publish Api App**

Go to publisher code -> rt.click EnterpriseService.Api project-> Publish🡪

Add an account (Email = <name>nmcop.com, use test credentials)

AppName= PublisherApiApp

Subscription= AzureLab1

Resource group= crm-lab(eastus)

App service plan= Crm-lab-integration-appserviceplan(S1, east US)

Publisher Url (after publishing)= http://publisherapiapp.azurewebsites.net/

**How to know/get publisher Url (which can be used in postman)**

Go to publisher code -> rt.click EnterpriseService.Api project-> Publish🡪

Summary/**SiteUrl**

**Procedure to test Logic App, Publisher Api App**

Ex: Merge LE plugin operation

Debugging publisher APi App using postman

1. Merge 2 contacts/accounts with same owner in CRM UI
2. Check if you get message in service bus explorer (as MergeEntity is registered under service endpoint in PI reg tool)
3. Get merge request from Run history of logic app – from scope/ParseJson/Output Body
4. Use this request as postman body as contact/account merge request (POST method)

This request will contain BaseLE under Entity/attributes and LE list (LE, Id src,life cycle code etc.) under EntityCollection for source/target.

Logical name (nm\_legalentityid), Guid etc.- outside Attributes(Entity collection) i.e, under Entities tag.

**How to parse json of base64encoded**

In parse json step **@json(decodeBase64(Content))** 🡪 code view= @json(decodeBase64(triggerBody()?['ContentData']))

**Logging using log4net**

*example*

private static readonly ILog log = LogManager.GetLogger(MethodBase.GetCurrentMethod().DeclaringType);

log.Info("Response Code:" + response.ResponseCode + ", Response Message:" + response.ResponseMessage);

**log4net configuration:-**

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?> |
| <!-- |
| For more information on how to configure your ASP.NET application, please visit |
| http://go.microsoft.com/fwlink/?LinkId=301879 |
| --> |
| <configuration> |
| <configSections> |
| <!-- For more information on Entity Framework configuration, visit http://go.microsoft.com/fwlink/?LinkID=237468 --> |
| <section name="entityFramework" type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFramework, Version=6.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" requirePermission="false" /> |
| <section name="log4net" type="log4net.Config.Log4NetConfigurationSectionHandler, log4net" /> |
| </configSections> |
| <log4net> |
| <!-- Define the output appenders --> |
| <appender name="RollingAppender" type="log4net.Appender.RollingFileAppender"> |
| <param name="File" value="C:\logs\HouseholdIntegrationService-Debug.log" /> |
| <param name="AppendToFile" value="true" /> |
| <param name="datePattern" value="yyyyMMdd" /> |
| <maxSizeRollBackups value="-1" /> |
| <maximumFileSize value="50MB" /> |
| <param name="RollingStyle" value="Composite" /> |
| <!-- Define the layout for this appender --> |
| <layout type="log4net.Layout.PatternLayout"> |
| <param name="ConversionPattern" value="%date %username [%thread] %-5level %class.%method - %message%newline" /> |
| </layout> |
| </appender> |
| <appender name="Statistics" type="log4net.Appender.FileAppender"> |
| <param name="File" value="C:\logs\HouseholdIntegrationService-Statistics.log" /> |
| <param name="AppendToFile" value="true" /> |
| <param name="datePattern" value="yyyyMMdd" /> |
| <maxSizeRollBackups value="-1" /> |
| <maximumFileSize value="50MB" /> |
| <param name="RollingStyle" value="Composite" /> |
| <!-- Define the layout for this appender --> |
| <layout type="log4net.Layout.PatternLayout"> |
| <param name="ConversionPattern" value="%date %username [%thread] %-5level %class.%method - %message%newline" /> |
| </layout> |
| </appender> |
| <!-- Setup the root category, add the appenders and set the default level --> |
| <root> |
| <level value="INFO" /> |
| <appender-ref ref="RollingAppender" /> |
| </root> |
| <logger name="Statistics" additivity="false"> |
| <level value="INFO" /> |
| <appender-ref ref="Statistics" /> |
| </logger> |
| </log4net> |
| <appSettings> |
| <add key="OrgServiceUri" value="https://crm7dnm.nml.com/CRM7DNM/XRMServices/2011/Organization.svc" /> |
| <add key="RunAsCurrentUser" value="false" /> |
| <add key="PublisherOptionSetPrefix" value="930750000" /> |
| <add key="PublisherOptionSetOffset" value="1" /> |
| <add key="MaxHouseholdFetchCount" value="25" /> |
| <add key="StatisticsLog" value="Statistics" /> |
| </appSettings> |
| <system.web> |
| <authentication mode="None" /> |
| <compilation debug="true" targetFramework="4.5.2" /> |
| <httpRuntime targetFramework="4.5" /> |
| </system.web> |
| <system.webServer> |
| <modules> |
| <remove name="FormsAuthentication" /> |
| </modules> |
| <handlers> |
| <remove name="ExtensionlessUrlHandler-Integrated-4.0" /> |
| <remove name="OPTIONSVerbHandler" /> |
| <remove name="TRACEVerbHandler" /> |
| <add name="ExtensionlessUrlHandler-Integrated-4.0" path="\*." verb="\*" type="System.Web.Handlers.TransferRequestHandler" preCondition="integratedMode,runtimeVersionv4.0" /> |
| </handlers> |
| </system.webServer> |
| <runtime> |
| <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1"> |
| <dependentAssembly> |
| <assemblyIdentity name="Microsoft.Owin" publicKeyToken="31bf3856ad364e35" /> |
| <bindingRedirect oldVersion="0.0.0.0-3.0.0.0" newVersion="3.0.0.0" /> |
| </dependentAssembly> |
| <dependentAssembly> |
| <assemblyIdentity name="Microsoft.Owin.Security.OAuth" publicKeyToken="31bf3856ad364e35" /> |
| <bindingRedirect oldVersion="0.0.0.0-3.0.0.0" newVersion="3.0.0.0" /> |
| </dependentAssembly> |
| <dependentAssembly> |
| <assemblyIdentity name="Microsoft.Owin.Security.Cookies" publicKeyToken="31bf3856ad364e35" /> |
| <bindingRedirect oldVersion="0.0.0.0-3.0.0.0" newVersion="3.0.0.0" /> |
| </dependentAssembly> |
| <dependentAssembly> |
| <assemblyIdentity name="Microsoft.Owin.Security" publicKeyToken="31bf3856ad364e35" /> |
| <bindingRedirect oldVersion="0.0.0.0-3.0.0.0" newVersion="3.0.0.0" /> |
| </dependentAssembly> |
| <dependentAssembly> |
| <assemblyIdentity name="Newtonsoft.Json" culture="neutral" publicKeyToken="30ad4fe6b2a6aeed" /> |
| <bindingRedirect oldVersion="0.0.0.0-6.0.0.0" newVersion="6.0.0.0" /> |
| </dependentAssembly> |
| <dependentAssembly> |
| <assemblyIdentity name="System.Web.Helpers" publicKeyToken="31bf3856ad364e35" /> |
| <bindingRedirect oldVersion="1.0.0.0-3.0.0.0" newVersion="3.0.0.0" /> |
| </dependentAssembly> |
| <dependentAssembly> |
| <assemblyIdentity name="System.Web.Mvc" publicKeyToken="31bf3856ad364e35" /> |
| <bindingRedirect oldVersion="1.0.0.0-5.2.2.0" newVersion="5.2.2.0" /> |
| </dependentAssembly> |
| <dependentAssembly> |
| <assemblyIdentity name="System.Net.Http" publicKeyToken="31bf3856ad364e35" /> |
| <bindingRedirect oldVersion="1.0.0.0-5.2.2.0" newVersion="5.2.2.0" /> |
| </dependentAssembly> |
| <dependentAssembly> |
| <assemblyIdentity name="System.Web.Optimization" publicKeyToken="31bf3856ad364e35" /> |
| <bindingRedirect oldVersion="1.0.0.0-1.1.0.0" newVersion="1.1.0.0" /> |
| </dependentAssembly> |
| <dependentAssembly> |
| <assemblyIdentity name="System.Web.WebPages" publicKeyToken="31bf3856ad364e35" /> |
| <bindingRedirect oldVersion="1.0.0.0-3.0.0.0" newVersion="3.0.0.0" /> |
| </dependentAssembly> |
| <dependentAssembly> |

<assemblyIdentity name="WebGrease" publicKeyToken="31bf3856ad364e35" />

<bindingRedirect oldVersion="0.0.0.0-1.5.2.14234" newVersion="1.5.2.14234" />

</dependentAssembly>

</assemblyBinding>

</runtime>

<entityFramework>

<defaultConnectionFactory type="System.Data.Entity.Infrastructure.LocalDbConnectionFactory, EntityFramework">

<parameters>

<parameter value="mssqllocaldb" />

</parameters>

</defaultConnectionFactory>

<providers>

<provider invariantName="System.Data.SqlClient" type="System.Data.Entity.SqlServer.SqlProviderServices, EntityFramework.SqlServer" />

</providers>

</entityFramework>

</configuration>

**Open Visual studio with different user (run as different user)**

Create a .bat file ex: Visual Studio\_RunAsdiffuser.bat with below command.

|  |
| --- |
| C:\Windows\System32\runas.exe /netonly /u:**nmtest\nai7033** "C:\Program Files (x86)\Microsoft Visual Studio 14.0\Common7\IDE\devenv.exe" |

If we run/open VS directly it takes windows credentials (nm/nai7033) but this crdential may not have access to some application/database but when we run with nmtest it will have the required access so we are running using different user.

**Nuget packages:-**

It can have Microsoft provided DLLs or custom/own DLLs.

We can refer them in our project using – Rt.click- references- Manage NuGet packages

Installed tab- Shows already installed DLLs.

Browse tab- Search for required Dll and install it.

\*we can update Dlls with old/new versions. Also we can uninstall existing Dlls.