GCcase

Development and Prototyping Directives

Note that this draft is a living document and additional directives and guidance will be created as the service is developed.

Consequently, it is critical that all clients coordinate with the GCcase Program Centre when planning the development of new capabilities using the case management product:

a. To avoid duplication of effort: capability may already be available or under development by another organization;
b. To avoid duplication of costs: other organizations may wish to join in the development and consequently share the costs.

Clients must abide by the Development and Prototyping Directives to ensure platform performance and stability. If the client does not abide by the Developing and Prototyping Directives, the GCcase Program Centre will not be responsible for problems resulting from an update, upgrade, patch or any other modification made to the GCcase base configuration. To maintain the integrity of the platform, in certain cases, GCcase may disable tenants that do not meet the directives outlined in this document.

GCcase

Table of Contents

1 Document Control Page

Document Number:	
File Storage Location:	**PSPC GCdocs**
Document Name:	Prototype and Development Directives
Date:	**March 2018**
Version:	**2.1**

2 Confidentiality

The information contained in this document is unclassified to the Government of Canada (GC).

3 Reference Documents

Please note: While some documents may reference CRM 201X, the guidelines still hold true for more recent versions of Microsoft Dynamics.

Software Development Kit for Microsoft Dynamics (D365)

https://msdn.microsoft.com/en-us/library/hh547453(v=crm.8).aspx

Use Microsoft Dynamics 365 services in code (D365):

Scalable Customization Design in Microsoft Dynamics (2016):

https://msdn.microsoft.com/en-us/library/gg309557(v=crm.8).aspx

Microsoft Dynamics Performance and Scalability Documentation (2016):

Scalable Customization Design in Microsoft Dynamics (2016):

Microsoft Dynamics CRM 2015 and Microsoft Dynamics CRM 2016 Performance and Scalability Documentation

Optimizing and maintaining the performance of a Microsoft Dynamics CRM server infrastructure (2011)

This white paper provides information that is designed to help readers achieve and maintain optimal performance of the server infrastructure supporting a Microsoft Dynamics CRM 2011-based business solution deployed in an on-premises or hosted environment.

https://www.microsoft.com/en-sg/download/details.aspx?id=27139

Implementation guide (D365)

This guide contains comprehensive information about how to plan, install, configure, customize, and maintain Microsoft Dynamics , an in-depth overview of all the components:

https://www.microsoft.com/en-us/download/details.aspx?id=50039

Best practices for developing with Microsoft Dynamics (D365):

The following best practices can help developers write code that performs better.

https://msdn.microsoft.com/en-us/library/gg509027(v=crm.8).aspx

Security (D365)

This page provides security information and best practices for the Microsoft Dynamics application, outlines minimum permissions required for installation and configuration:

http://msdn.microsoft.com/en-us/library/hh699825.aspx

Administration (D365)

Microsoft Dynamics administration best practices, setting up user roles and administrative permissions:

http://msdn.microsoft.com/en-us/library/hh699744.aspx

Alternatives to direct reporting server access (D365)

Direct access to SSRS and SSIS are not supported, however report definition (.rdl) files can be used.

https://technet.microsoft.com/en-us/library/dn531151.aspx

Supported web browsers (D365)

Users can access the Microsoft Dynamics Web application on the most recent versions of the following browsers:
https://technet.microsoft.com/en-us/library/dn531055.aspx

Software Development Kit (D365)

The complete software development kit for Microsoft Dynamics CRM, information for developers writing server side code, custom business logic, integration modules, workflow assemblies, and plug-ins:

https://www.microsoft.com/en-ca/download/details.aspx?id=50032

Plug-ins (2016):

Security context for CRM plugins.

https://msdn.microsoft.com/en-us/library/gg334752(v=crm.8).aspx

Supported extensions for Microsoft Dynamics (D365):

Customizations made using methods other than those described in the following document are unsupported and could cause problems during updates and upgrades to Microsoft Dynamics.

https://msdn.microsoft.com/en-us/library/gg328350.aspx?f=255&MSPPError=-2147217396#Unsupported

CRM Solution Lifecycle Management (2011)

CRM Solution Lifecycle Management – component reuse contains examples on managing development efforts:

http://www.microsoft.com/en-ca/download/details.aspx?id=39044

Microsoft Dynamics Custom Code Validation Tool (2015)

The tool will flag items that are unsupported.

https://www.microsoft.com/en-us/download/details.aspx?id=45535

Microsoft Dynamics Server Best Practices Analyzer (2015)

The tool will flag validate security roles and permissions:

https://www.microsoft.com/en-us/download/details.aspx?id=48268

Optimizing client performance

Optimizing and maintaining client performance for Microsoft Dynamics:

(2011)

http://www.microsoft.com/en-ca/download/details.aspx?id=23261

(2016)

https://mbs.microsoft.com/customersource/Global/CRM/learning/documentation/user-guides/PerformanceOptimizationsCRMOnlineSuccess

Optimize form performance

Forms that load slowly can reduce productivity and user adoption.

https://technet.microsoft.com/en-us/library/dn531124.aspx

Optimize view performance

Dynamics CRM Quick Find Performance & Records per Page Setting **(2011)**

https://blogs.msdn.microsoft.com/emeadcrmsupport/2014/07/08/dynamics-crm-2011-quick-find-performance-records-per-page-setting/

GCcase

Onboarding

This document is designed to provide information to client organizations wishing to join the Shared Case Management Solution (GCcase) Program. It will provide a detailed description of the onboarding process in terms of activities to be performed and what is required of the client

https://gcconnex.gc.ca/file/group/4087274/all#16497202

Email Configuration

Configuring email integration.

GCcase Email Configuration - Best Practices https://gcconnex.gc.ca/file/download/24894730

Track Outlook email by moving it to a tracked Exchange folder

Track customer interactions from any location, and from virtually any device by using folder tracking.

https://docs.microsoft.com/en-us/dynamics365/customer-engagement/admin/track-outlook-email-by-moving-it-tracked-exchange-folder

Microsoft Exchange Integration Guide

Integrate (synchronize) an email system with Microsoft Dynamics

Set up Dynamics for Outlook

Setting up outlook to work with Dynamics CRM

https://www.microsoft.com/en-us/dynamics/crm-customer-center/set-up-dynamics-365-for-outlook.aspx

Help & Training: Learn more about Dynamics App for Outlook

Exchange Impersonation

Permissions required for exchange integration

Reference document: Impersonation guide

Plug-in isolation, trusts, and statistics

“Sandbox mode” is the **only** supported execution environment for plug-ins as it is more secure, supports run-time monitoring and statistics reporting.

https://msdn.microsoft.com/en-us/library/gg334752.aspx

Single Tenant vs Multiple Tenants

Using multi-tenancy in Microsoft Dynamics CRM to address challenges in enterprise business environments

https://gcconnex.gc.ca/file/view/6720744/guidelines-for-single-vs-multiple-tenancies

https://www.microsoft.com/en-us/download/details.aspx?id=36056

GCcase

Supported Browsers (D365)

https://technet.microsoft.com/en-us/library/hh699710.aspx

Supported Workstation (D365)

https://technet.microsoft.com/en-us/library/hh699710.aspx

Microsoft Dynamics deployment-level tracing

In Microsoft Dynamics, create trace files that monitor the actions that are performed by the server and client applications. Trace files are helpful to troubleshoot error messages or other issues in Microsoft Dynamics.

https://technet.microsoft.com/en-us/library/hh699694(v=crm.8).aspx

Data Import

Import data into Microsoft Dynamics 365 by using the data import feature. Data import lets one upload data from various customer relationship management systems and data sources into Microsoft Dynamics. Data can be imported into standard and customized attributes of most business and custom entities. Related data, such as notes and attachments can also be included.

https://msdn.microsoft.com/en-us/library/gg328321.aspx

4 Purpose

The GCcase Program Centre's goal is to provide a shared case management platform that is stable, current with the latest technology, and offers a flexible, configurable solution that can be leveraged across the GC. Consequently, it is essential that the core underlying solution remain within the GCcase Program Centre’s supported architecture. The following guide attempts to provide GCcase clients with best practices surrounding solution development in alignment with the common solution for the GC.

A core element of this mandate is to leverage and share solution configurations, workflows and integration points among clients. There are many benefits which include: reduced development time, use of best practices, knowledge transfer, and reduced costs. The GCcase Program Centre will endeavor to keep any administrative process surrounding solution development to a minimum. In turn, all clients will benefit from the work effort of the entire community.

The **Microsoft Dynamics Implementation** Guide (D365) is a complete planning, installation, administration, customization, operating and maintenance guide available from Microsoft.

5 Contact the GCcase Program Centre First

Before undertaking any development or configuration activity, please contact the GCcase Program Centre. It will only take a few moments to determine if there are resources or information that can be leveraged to add work to the program catalog.

6 Sharing

The GCcase Program Centre's goal is to provide all partners with the ability to leverage any work performed by the community. To facilitate this, the GCcase Program Centre will implement a reporting procedure to track activities across all partners.

7 Security

To secure the application, the GCcase Program must undergo an assessment of security controls as stipulated by the IT Security Directorate (ITSD) within PSPC.

All controls identified by ITSD will be assessed and validated and include such categories as:

- Access control and its sub-controls, Account Management, Access Enforcement, Separation of Duties and Least Privilege,
- Configuration Management,
- Identification and Authentication,
- Incident Response, and
- Risk Assessment

SSC conducts periodic security scans across its hosted infrastructure looking for vulnerabilities. These vulnerabilities are addressed by SSC or flagged to the platform for action.

The product itself is configured and installed using the principle of least privilege.

Review and adhere to the following recommendations provided in the reference documents:

- Security Considerations for Microsoft Dynamics
- Microsoft Dynamics Administration Best Practices

8 Right to Recourse

To maintain platform stability and performance, the GCcase Program Centre reserves the right to disable any solutions adversely affecting the platform and/or other client solutions.

Some Examples:

1. Quick Find column count is excessive
2. Dashboard visualizations are excessive
3. Columns and grids are excessive
4. Incorrect or missing search indexes
5. Searches are not optimized to return a reasonable set of records
6. The Microsoft Dynamics Custom Code Validation Tool reports unsupported operations
7. Sudden significant increase of a tenant size that risks exceeding total capacity of allocated server space, risking platform stability and other hosted tenants

9 Onboarding

This following document is designed to provide information to client organizations wishing to join the Shared Case Management Solution (GCcase) Program. It will provide a detailed description of the onboarding process in terms of activities to be performed and what is required of the client.

Onboarding Documentation

10 Key Directives

10.1 *General*

The following are the general principles that govern the Solution development and design methodology:

- Performance integrity for all outweighs individual client's solution availability
- No Dynamics activities, either core platform or implementation, work completely in isolation
- All Dynamics activities interact with the same database resources, either at a data level or an infrastructure level such as processor, memory, or IO usage
- Dynamics system constraints are in place to prevent an excessively long running action from affecting other users or system performance. These default constraints will not be modified.
- Running into a constraint is an indicator of a design problem and not a platform problem

10.2 *Key Directives and Best Practices*

The following are the key *directives* from the GCcase program team:

1. It is generally a much better strategy to start with a small solution to get acclimated to the application lifecycle of a Dynamics CRM project.
2. Ensure that every team member is VERY familiar with the following document:
    - To ensure that client solutions are responsive and error free:
        - **Microsoft Dynamics Performance and Scalability Documentation (2016)**
    - "Sandbox mode" is the **only** supported execution environment for plug-ins as it is more secure, supports run-time monitoring and statistics reporting.
        - **Plug-in isolation, trusts, and statistics**
3. No third party JavaScript libraries such as jQuery
4. The GCcase platform is not a document repository, please consult the GCcase program office if storing documents in CRM is desired.
5. Consult the MS Best practices for developing with Microsoft Dynamics documentation:
    - **Best practices for developing with Microsoft Dynamics (D365):**
6. Only utilize the provided object model Xrm.Page for scripting
7. Direct access to database is not supported. There are security implications and access to DB can nullify the security layer within Dynamics. This includes access to:
    - Reporting Services (SSRS): **Alternatives to direct reporting server access (D365)**
    - SQL Server Integration Services (SSIS): Access to the SSIS functionality is not supported.
8. The use of application programming interfaces (APIs) other than the documented APIs in the Web services CrmDeploymentService, CrmDiscoveryService, CrmService and MetadataService is not supported.
    - For more information see the following: Supported extensions for Microsoft Dynamics (D365)

9. Database scripts cannot be run against the GCcase database.
    - GCcase does not support execution of client SQL scripts
10. Third party applications are not supported unless approved by the GCcase Program Centre; basic mandatory requirements below must be satisfied:
    - The platform is not impacted; and
    - It doesn't require GCcase program team to install any new components or modify existing ones.
11. The Microsoft Dynamics Custom Code Validation Tool must be utilized to correct any unsupported operations to ensure that all client solutions are responsive and error free.
12. The program centre will not adjust the following parameters:
    - Plug-In timeouts
    - SQL Timeouts
    - Workflow limits
    - Maximum concurrent connections
    - Execute Multiple
13. Perform transaction diagnosis and identify lock chains locally: the GCcase Program Centre will not perform transaction diagnosis except to identify issue sources; should issue be identified it is the client's responsibility to correct and modify the lock chains.

Apart from the directives, the following are the list of ***suggestions*** for the tenant's solution design and coding team:

14. Design for ExecuteMultiple = 2.
    - The processing of large groups of these requests can tie up vital resources in CRM at the expense of end user requests, therefore this is limited to 2 concurrent ExecuteMultiple requests per organization
    - Please see below for more details.
15. Security
    - Teams Setup:
        - Always add users in the same order: avoid deadlocks
        - Only update users if they need to be updated: avoid invalidating users' caches unnecessarily
    - Owner vs. access teams
        - If users' teams change regularly, be careful about using owner teams heavily; every time they change they invalidate the user's cache in the web server
        - Ideally make changes when the user isn't working, reduce impact, such as overnight
    - Lots of team memberships/Bus
        - Consider carefully scenarios with lots of teams/Bus; it adds to the complexity of the security processing
    - Cascading behaviour
        - Consider cascading sharing, for example: assignment.
    - Careful updating of user records
        - Don't regularly update system user records unless something fundamental has changed as this forces the user cache to be reloaded and the security privileges to be recalculated, an expensive activity
        - Don't use system user to record how many open activities of a user

The following are the coding best practices for Workflows and Plugins which may assist in reducing the volume of unhandled exceptions on GCcase platform:

- If a workflow is triggered on an event of record status change, the record attribute updates must be conditional to avoid the cases when the record is inactive, disabled or cancelled.
- When a GUID of a record is obtained programmatically (e.g. from a property of another record), ensure that a record with the GUID exists before commit any record update or removal.

- Other best practices for developing with Microsoft Dynamics are listed:
  - **Best practices for developing with Microsoft Dynamics (D365):**

11 Planning

11.1 *General considerations*

It is generally a much better strategy to start with a small solution to get acclimated to the application lifecycle of a Dynamics CRM project.

Planning for Microsoft Dynamics, like any enterprise-wide software, is a significant task. A team of people is required for planning Microsoft Dynamics. Several roles may be filled by one person. In larger organizations, each role may be divided among several people. These roles include, but are not limited to, the following:

- Business managers- Responsible for determining how the business will use Microsoft Dynamics. This includes mapping processes to Microsoft Dynamics , deciding on default values, and identifying any required customizations or integrations
- Customization technical staff- Responsible for implementing the planned customizations
- Network technical staff- Responsible for determining how connections will be made to the shared service and how users will access the system
- Project manager- Responsible for managing an enterprise-wide implementation project
- Integration specialist- Responsible for implementing line of business integration

11.2 *Single or Multiple Tenants*

Organizations may require multiple tenancies if the following are required:

- Volume: data needs are expected to exceed 250GB-500GB. Please note this does not include attachments and document stored in a document management system.
  - The GCcase platform is not a document repository, please consult the GCcase program office if storing documents in CRM is desired.
- Disparate requirements for key entities: a set of processes related to help desk and an unrelated set of processes related to G&C's all attempting to make use of the case entity.

An organization may require a single tenancy if the following are required:

- Out of the box reporting across all processes.

Reference document: **Single Tenant vs Multiple Tenants**

11.3 GCcase Authentication

GCcase utilizes claims-based authentication, an identity access solution designed to provide simplified user access and single sign-on access to Microsoft Dynamics CRM data and other applications hosted in the SSC end state datacenters.

The SSC authentication solution for GCcase is Active Directory Federation Services and Active Directory.

Departmental Active Directory credentials are copied from the departmental Active Directory into the SSC end state datacenter active directory. This synchronization of credentials is a pre-requisite to onboarding to the GCcase platform. It is addressed at length during the onboarding process. **Onboarding Documentation**

Users can authenticate to their GCcase tenants via ADFS by entering their departmental Active Directory credentials providing a CRM user has been created for that user. The administrative user is created by the GCcase program during onboarding. The client administrative user is then responsible for create additional users.

11.3.1 Other Departmental Users

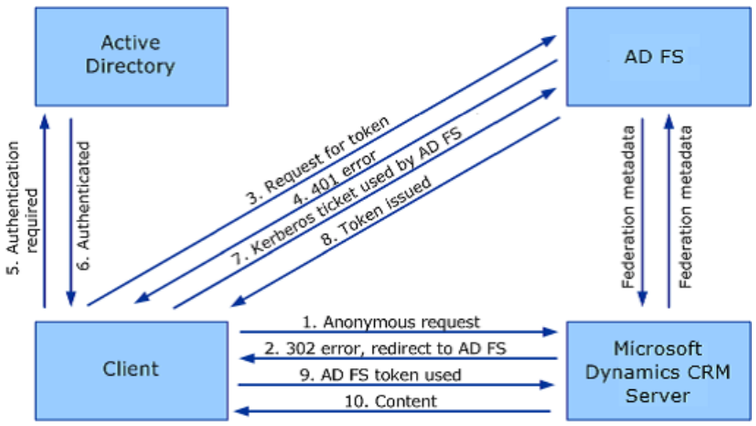
The client administrative user can create users with the sole limitation that the user credentials be present in the SSC End State Datacenter Active directory instance. (I.e. users have been “synced”). Therefore client administrators can create GCcase CRM accounts for users in other departments to support and requirements to do so.

11.3.2 ADFS flows explanation

Claims-based authentication is built on Windows Identity Foundation (WIF), a framework for building claims-aware applications and security token service (STS) that is standards-based and interoperable. Interoperability is provided through reliance on industry standard protocols.

In claims-based authentication, an identity provider, or security token service, responds to authentication requests and issues SAML security tokens that include any number of claims about a user, such as a user name and groups the user belongs to. A relying party application receives the SAML token and uses the claims inside to decide whether to grant the client access to the requested resource. Claims-based authentication can be used to authenticate the organization’s internal users, external users, and users from partner organizations.

The following diagram explains information flow between the different authentication components in the GCcase environment:



11.3.3 My Key and GCKey

GCcase does not currently support MyKey or GCKey directly. These solutions can, however, be utilized when interacting with a GCcase indirectly, for example a client portal could be setup that interacts with GCcase via an active directory account while the access to the portal could be controlled by MyKey or GCKey. Please note that these activities are out of scope for GCcase.

11.4 *Managed vs unmanaged solutions*

See: CRM Solution Lifecycle Management section

11.5 *Required Client Infrastructure*

This section details on the two key aspects of client infrastructure – workstation and network.

11.5.1 Workstation

Later versions of Internet Explorer provide enhancements that improve page rendering. It is recommended that computers running Microsoft Dynamics clients be installed with the latest version of Internet Explorer supported within a specific business environment. For an optimal experience with Microsoft Dynamics, use the latest version of Internet Explorer. Recent versions of Internet Explorer includes JavaScript acceleration functionality to help improve client-side performance.

The following are the key components on a client workstation.

11.5.1.1 Web Clients

11.5.1.2 Supported Configurations Web Browsers and Mobile Devices

Operating Systems	Windows 8 Pro and RT	Windows 7	Windows Vista	Apple OS X10.8	IOS 6 and above (iPad)	Android 4.2.2 (Nexus 10)
Browser	• IE 10 desktop • IE 10 Modern • Latest version of Chrome • Latest version of Firefox	• IE 8 • IE 9 • IE 10 desktop • Latest version of Chrome • Latest version of Firefox	• IE 8 • IE 9 • Latest version of Chrome • Latest version of Firefox	• Latest version of Safari	• Latest version of Safari	• Chrome

Not supported:

- IE7 on On-Premise
- Windows XP
- Outlook clients using IE7 or Win XP

For latest updated information, please refer to: **Supported Browsers (D365)**

11.5.1.3 Power

Microsoft Dynamics performs best when a computer's power plan is set to "High Performance," which is the recommended setting to ensure the fastest performing computer for running Microsoft Dynamics for Outlook and Microsoft Dynamics for Outlook with Offline Access, and the Microsoft Dynamics Web client. For more information: **Supported Workstation (D365)**

11.5.1.4 Hardware

For client computers with hardware at or near minimal-level requirements, consider turning off non-critical business processes, video streaming, and software such as games and music programs, to accelerate performance. Because this issue can have a greater impact on client computers configured for offline access, third-party add-ins for Microsoft Office Outlook can also be disabled to improve the performance of Microsoft Dynamics for Outlook. For more information: **Optimizing client performance**

11.5.1.5 Virus Scan

Microsoft Dynamics client performance can also be affected by desktop security software, including malware applications and antivirus programs, which can lock certain files and make them inaccessible to other applications. In addition, some of these software packages include a feature called *ScriptScan*, which scans all web pages for malicious script and is known to affect the performance of web-based applications such as Microsoft Dynamics CRM.

Most of these programs provide the ability to disable scanning on specific web sites, and if so, adding the URL of the Microsoft Dynamics organization to a list of excluded sites can help to improve performance in these situations. For more information: **Optimizing client performance**

11.5.2 Network

The primary characteristics of a network that affect the performance of Microsoft Dynamics clients are bandwidth and latency.

- Bandwidth is the width or capacity of a specific communications channel
- Latency is the time required for a signal to travel from one point on a network to another

One of **the main causes of poor performance of Microsoft Dynamics** clients is the latency of the network over which the clients connect to the Microsoft Dynamics solution. Lower latencies (measured in milliseconds) generally provide better levels of performance.

Note that even if the latency of a network connection is low, bandwidth can become a factor if there are a lot of resources sharing the network connection, for example to download files and send and receive email.

The high bandwidth of a network does not guarantee low latency. For example, a network path traversing a satellite link often has high latency, even though throughput is very high. It is not uncommon for a network round trip traversing a satellite link to have five or more seconds of latency. An application designed to send a request, wait for a reply, send another request, wait for another reply, and so on, will wait at least five seconds for each packet exchange, regardless of the speed of the server.

It is recommended to test the client performance with potential bandwidth or latency issues. For example, users connecting to a standard configuration of Microsoft Dynamics in a 10ms latency environment can expect Account or Contact forms to load within 2 to 2.5 seconds, while users connecting to a similar configuration of Microsoft Dynamics in a 150ms latency environment can expect 3 to 3.5 second load times for the same forms.

Microsoft Dynamics is designed to work best over networks with latency under 150 ms (milliseconds) and bandwidth greater than 50 KBps (kilobyte per second).

For more information: **Optimizing client performance**

12 CRM Solution Lifecycle Management

This section gives an overview of a typical solution life cycle.

12.1 *CRM Solution Concepts*

Customizers and developers use solutions to author, package, and maintain units of software that extend Dynamics. Solutions are distributed to organizations so that they can use Dynamics to install and uninstall the business functionality defined by the Solution.

Solutions and layers are fundamental concepts within CRM that need to be fully appreciated and understood to construct an approach to lifecycle management for Dynamics applications. There are three key concepts that need to be introduced:

- Solution Packages: Act as containers for functionality required to be deployed as a unit

- Layers: The consequence of a specific component being affected by change within one or more solutions
- Managed Properties: The mechanism to control how layers interact with each other
    - Configuration of the level of lock down; controlling the degree of further customization that is possible to components within a given solution package

12.2 *Solution Management Processes*

To create the components of a CRM solution, it is necessary to work within the unmanaged layer. In solution terms this is analogous to working with source code. Continuing the analogy, source code would be compiled into binary form prior to deployment to production, and this is similar to the export of a Managed Solution.

From an enterprise development perspective, it is beneficial to test the deployment process for production as early and as often as possible.

12.3 *The Lifecycle from a CRM Solution Perspective*

An application for Dynamics consists of one or more Solution Packages that will be deployed into a CRM organization. The deployment processes (initial version, hot fixing, next version) need to be repeatable and predictable. From a CRM Solution perspective, these processes must be managed irrespective of whether Source Code Control may support the approach.

Note: Maintaining a complex range of solutions, and the dependencies between them, is an important role the GCcase Program Centre will undertake. This can consume a significant amount of time which emphasizes the need for the Program Centre to be current with efforts across the partners.

12.4 *Production Migration*

As part of production migration, developers should ensure:

- All custom developed plugins are registered in Isolated Mode.
- No exceptions in Workflows with Custom Activities.
- No exceptions in custom developed Plug-ins.

The following gating process and/or check list is applicable during migration of the first tenant solution from CDT to Preproduction and Production environments:

- The department confirms that the Solution is developed and tested; GCcase technical team validates the tenant logs using the logs described in *Section 13.1 Monitoring*.
- Issues, if found, will be identified and sent to the tenant by the GCcase team.
- The department sends back a report on how the issues will be addressed.
- Once the tenant confirms that the issues are fixed and re-tested, GCcase technical team will validate the logs.
- The client gets a “Go” to move its first solution from CDT to Pre-Prod and Production, if no new issues are registered.

12.5 *Unmanaged Vs. Managed Solutions*

Although Dynamics can be customized directly, it is essential that all customizations occur within the context of a solution. Customizing the CRM system directly uses a special solution called the Default Solution. The Default Solution contains all the components in the system.

At no point should the Default Solution be updated, this includes avoiding the use of the “Form Editor” and “Edit Process” options on the “More Commands” menu on each entity form. All configurations should be made in a solution through the Settings > Solutions area.

Guidance on the use of solutions and application lifecycle for development are well documented in the “ALM for Microsoft Dynamics: CRM Solution Lifecycle Management”, at the time of publishing this document, no recent version of this document was available. However, the material still applies.

13 Solution Development and Configuration

This section details into the best practices and key directives on solution development and configuration.

13.1 *General*

See: Key Directives

13.2 *Forms*

13.2.1 What is a form?

- A Form is the main display mechanism for an entity
- The key UI component that is used to create, view or edit entity records
- Multiple types of forms available
    - Main, Mobile, Quick Create, Quick View

13.2.2 Why Form design is important?

- From an end user’s perspective the application should be responsive enough to do the job
- If a form takes over say 5 seconds to load the user may begin to lose interest
- Frustration will kick in if the form lags, especially for repetitive tasks.

13.2.3 Form Design Recommendations

- Limit the number of fields on the form
    - There is no way to measure field loading time, but the less content on the form, the better it will perform
- Role-based forms
    - Should be used to create simplified forms for users rather than jamming everything in a single form and then try to control logic (show/hide sections, tabs, attributed, etc. based on user roles) in the code
- Collapse Tabs
    - Try and use collapsed tabs where possible to have better load performance
    - Iframe and subgrids should be loaded through collapsed tabs where possible
        - The main benefit of this is that the content is not loaded until the tab is expanded vs on the initial load stressing the form
- Be mindful when attaching JavaScript libraries to the form

- Avoid loading unnecessary scripts on the form as downloading scripts take time. Though this gets cached and is not downloaded again till the cache is cleared, it's still a good practice to remove the unused libraries
- Reduce the # of calls between the client and server

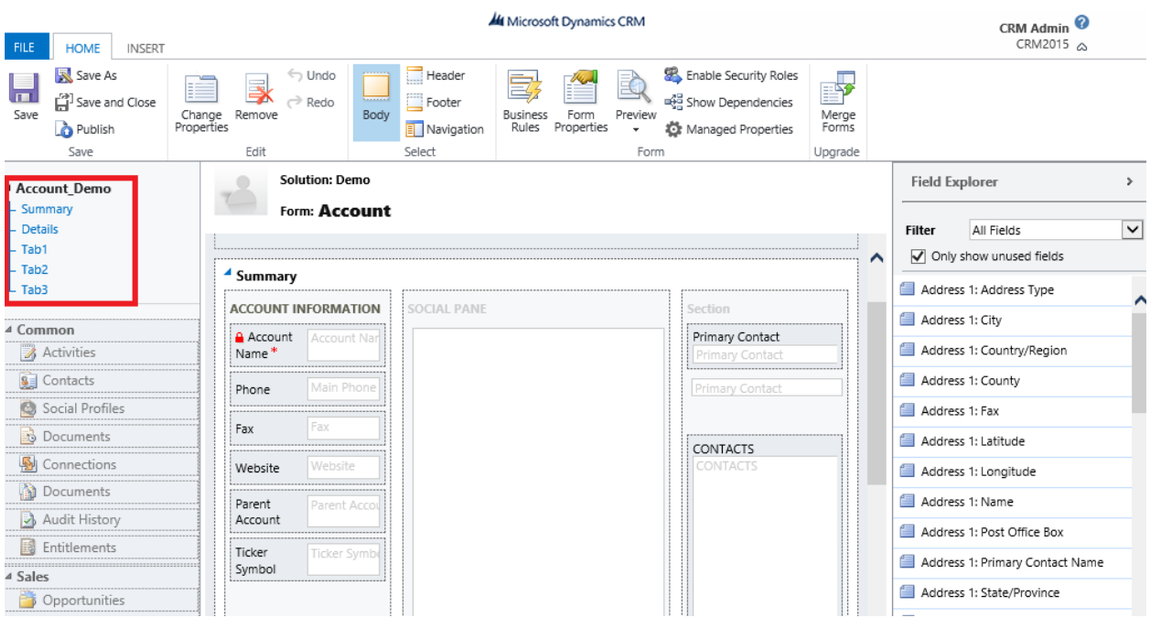
References: Optimize form performance

13.2.4 Example(s) from the Field

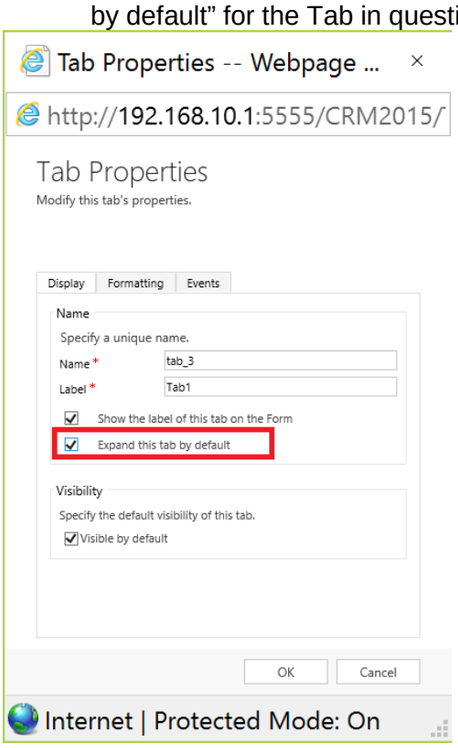
- Client with a form performance issue
  - Key points
    - Heavily customized form
    - Lot of attributes on the form
    - Lot of logic triggered via JavaScript
    - Expanded tabs
    - Form load time around 6-7 secs
  - After following some of the recommendations listed here, the form load time were brought down to 3-4 secs
  - Tools used: Fiddler, CRM Form Performance (CTRL+SHIFT+Q), Solution Review
- Another example where a form was taking approximately 40 secs to load:
  - After reviewing , man Web Service calls were found
  - Optimization and removal of many web service calls, brought the form load down to around 10 secs (there is still room for improvements though)
  - Tools used: Fiddler & Solution Review

13.2.5 Example(s) from the Field - Improving form performance

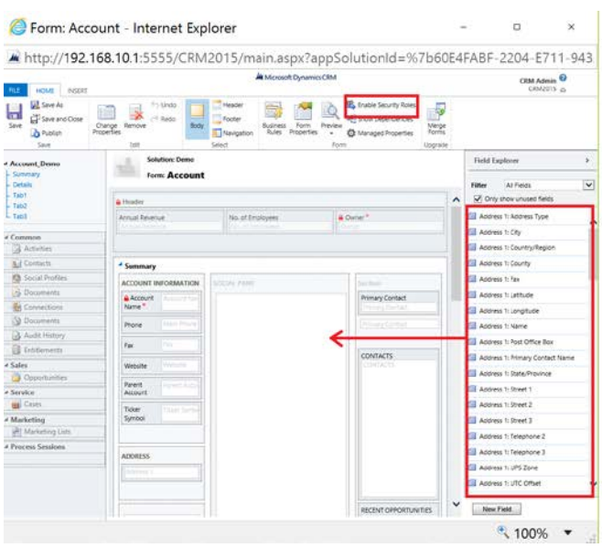
- Below is a screenshot of a demo account form where there are 5 tabs all expanded by default

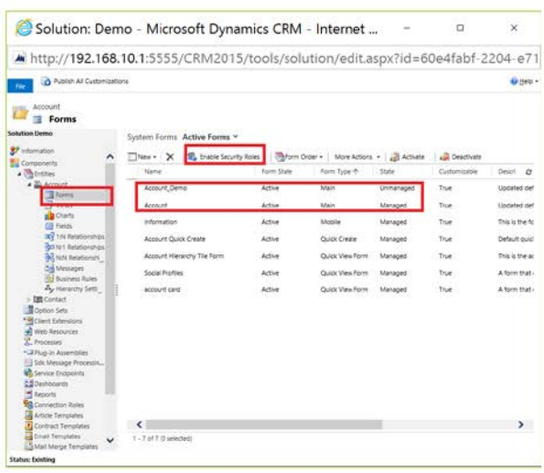


- In the form designer this behavior can be changed by unchecking the “Expand this tab by default” for the Tab in question



- To better manage the forms, the number of fields on any tab/section, security roles the solution designer should be used; below is the snapshot highlighting some of the key areas





13.3 *Views*

13.3.1 What is a view?

- Entity Views are saved queries that retrieve and display data(entity records)
- Views are the most extensively used UI component in the application (public views being the landing for all entities)
- Multiple types of views each with its own significance
    - Public (1 of these is set as Default)
    - Advanced Find
    - Associated
    - Quick Find
    - Lookup

13.3.2 Why view design is important?

- The importance is similar to the one discussed for the forms above, views are a key UI component i.e.
    - It should be responsive
    - Should be very responsive to avoid user frustration.

13.3.3 View Design Recommendations

- Limit the number of fields in the view definition
    - There is no predefined count for the number of attributes that can appear on the view, but it is recommended to keep it to the minimum for the task at hand
    - In many instances developers add attributes that are not required.
- Limit the number of rows returned
    - Similar to controlling the number of attributes on the view, filters should be added to views to limit the results
- Quick Find Views
    - Keep the find columns to a minimum (3-4). The more find columns, the heavier the underlying query becomes resulting in poor performance
- Since end users can create their own personalized views training should be provided to ensure users follow the above guidance

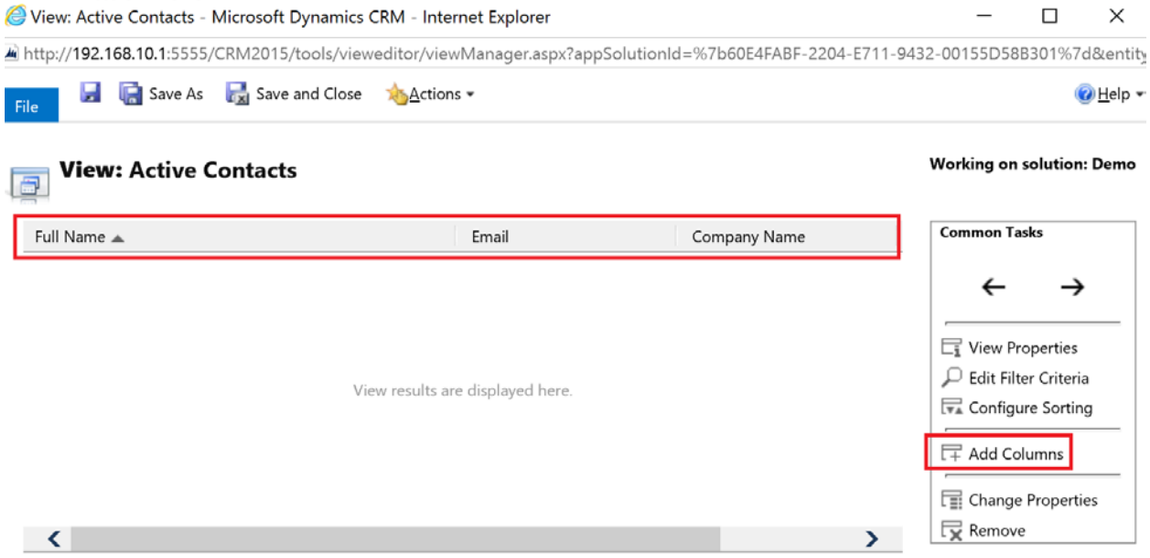
References: Optimize view performance

13.3.4 Example(s) from the Field

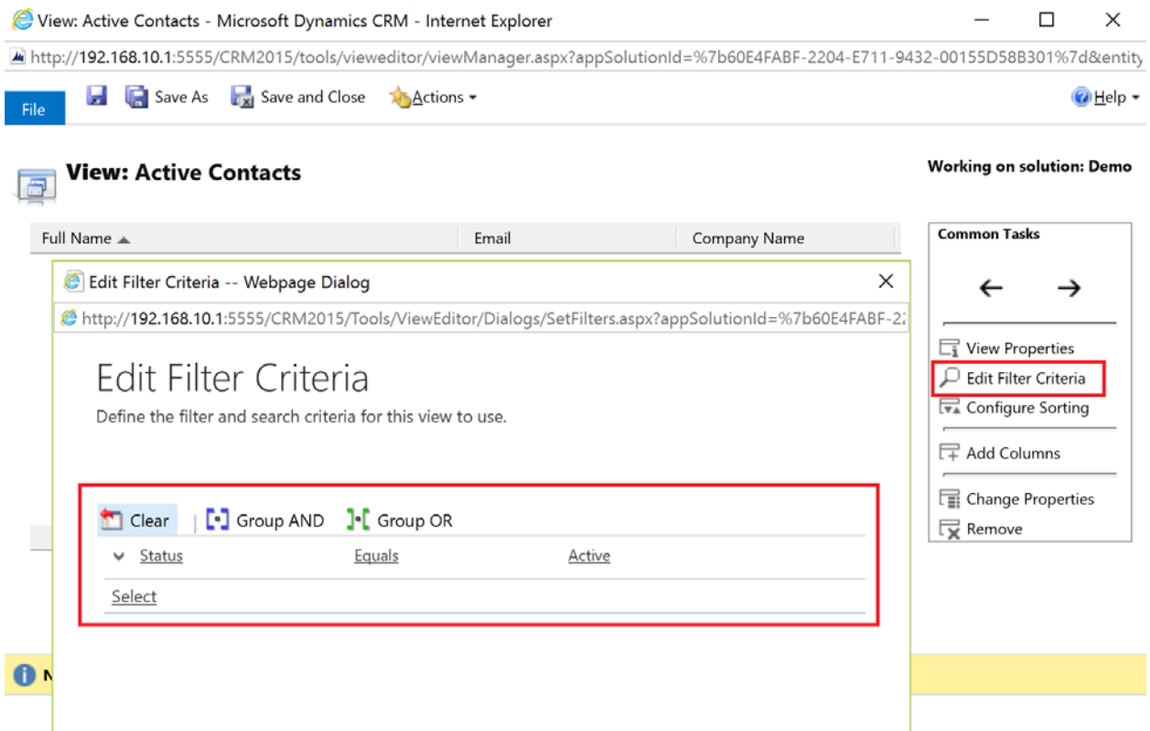
- A slow search when using Quick Find for Contact records
    - Key points
        - Around 12 “find” columns
        - Some custom attributes added to search like SIN #
        - Out of the 12 find columns there were 3 redundant column
            - Firstname, lastname, middlename and fullname all added as search column, while only the fullname was required
        - Search was taking approximately 30 secs
    - After following some of the recommendations outlined, the search was brought down to less than 1 sec (i.e. reduced the find column in the view)
    - Tools used: Solution Review

13.3.5 Example(s) from the Field - Improving view performance

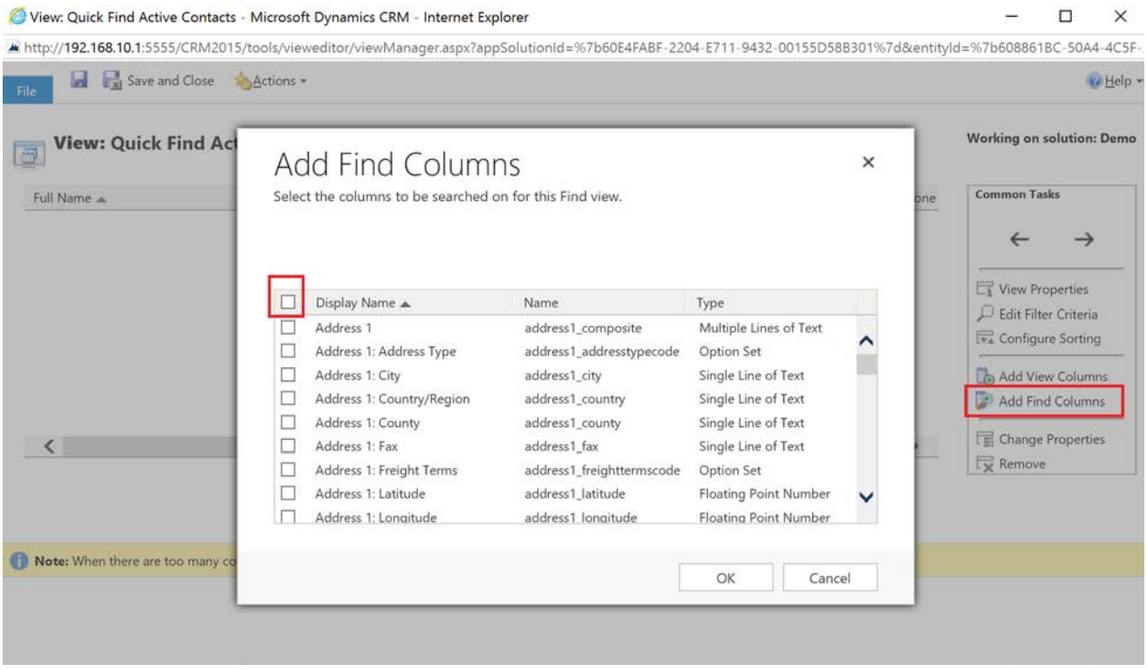
- Below is a screenshot of an active contact view showing the columns added and how to add more



- To add filter click on “Edit Filter Criteria” and set the filter conditions



- For Quick Find view click on “Add Find Columns” and review/add/remove search attributes



13.4 *Dashboard*

13.4.1 What is a dashboard?

- Dashboards collect and present an organization’s most important information in a single place
- They provide insight into actionable business data viewable across the organization
- 2 types of dashboards are available:
  - System: created by system administrator viewable across the organization
  - User: created by individual user(s) that can be shared with other user(s)
- They can be composed of:
  - Charts
  - Lists
  - Web Resources
  - IFrames

13.4.2 Why dashboard design is important?

- Dashboard are a key UI component within the Dynamics CRM application and are often used to display KPIs(Key Performance Indicators)
  - Dashboards capture a lot of data that are loaded separately in parallel connections
  - It is crucial that it loads quickly

13.4.3 Dashboard Design Recommendations

- Limit the data (both at a column level and pre-filtering level) in reference to the subgrids/lists put on the Dashboard
- Remove redundant items
  - Many times there will be the same dataset on the dashboard but with slightly different filter, ex. Active Contacts & Inactive Contacts
  - If the datasets are very similar, provide options for the users to switch the view as needed as opposed to loading both datasets

- Keep it visual in the form of charts, etc. (as the aggregate information performs better than raw data)
- Try to minimize the component count to 6 (as there is an option to create a dashboard with more than 6 components via SDK)
- Keep in mind if there are more than 6 components only the 6 will be loaded initially rest of them will be on demand

13.5 *Reporting*

13.5.1 Reporting Background

- Reports in Dynamics CRM are based on SSRS (SQL Server Reporting Services) engine
  - CRM Reports are stored in SSRS
  - A copy also gets copied in the Reports entity in the application
- Reports are used to provide useful business information to the users
- Reports rely on SSRS
  - FetchXML based: uses Microsoft Dynamics Report Authoring Extension that are used to retrieve data using FetchXML

13.5.2 Why reporting design is important?

- Report requirements cannot be generalized as they are all different some can be quite simple and some can be complex
- Simple reports are usually not an issue
- Complex reports can impact the system and the user experience
  - Slow running reports
  - Timeout issues
  - Added stress on the SQL server that may affect the overall performance of the application

13.5.3 Reporting Design Recommendations

- Review the number of fields returned in each dataset, always retrieve what is required for the report vs getting all columns
- Ensure unused datasets in the report definition are removed, as all the datasets on the report get executed regardless of whether they are used
- Use “No lock” hint
- Add filters and parameters to reduce the data retrieval
- Use GROUP BY judiciously: try to implement Grouping at the dataset level vs the report as it will help reduce the data transfer and processing on SSRS engine

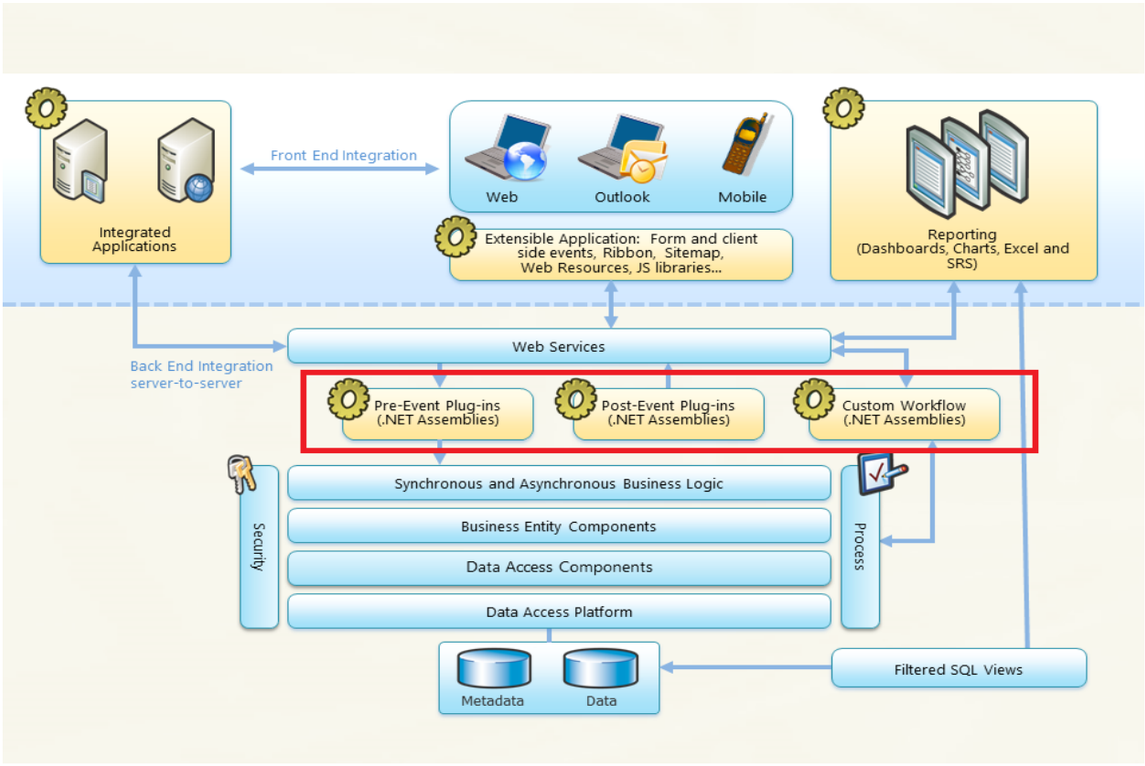
13.5.4 Example(s) from the Field

- Client with a slow report issue
  - Key points
    - Report run monthly
    - Initially report was fast but with time it slowed down
    - Taking approximately 45 secs to execute and render
    - Though not the entire story, the issue was primarily related to the report design, so changes were focussed on removing unused datasets in the report which brought down the report execution to 10 sec
  - Tools used: Report Design Review

13.6 *Server Side Code*

13.6.1 Background

- Dynamics CRM has multiple extensibility points that allows the application to be extended to meet business requirements. The ones highlighted in red are server side components



- Dynamics CRM has 2 extensibility points related to server side code:
  - Plug-ins: custom code to modify standard platform behaviors
  - Custom Workflows: custom code incorporated into workflow rule definition
- Both plug-in and custom workflow activity compiles into a .Net assembly.
- Plug-in code attaches to event execution pipeline which has multiple stages as below
  - 10 – Pre-Validation
  - 20 – Pre-Operation
  - 30 – Platform Core Operation
  - 40 – Post-Operation
- Since both are written in .Net they should follow .Net coding standards in general.

13.6.2 Why design is important?

- A lot of heavy lifting including platform level operations are handled by the Dynamics CRM server
- Since the custom code operates in the platform's event execution pipeline, bad code/practices can affect the entire platforms performance.
- Always ensure that only supported practices are used, refer to the documentation/samples from SDK.

13.6.3 Server Side Code Recommendations

- Limit data retrieval: Retrieve the minimum amount of data that the application needs by specifying the column set, which is the set of entity attributes to retrieve. Also restrict the amount of data returned by using appropriate filters.

- Reduce the retrieve calls by using context: There is no need to perform a Retrieve operation on the primary entity targeted by the platform event. Developers should instead leverage the context mechanisms such as Input/OutputParameters and Pre/Post Event Images to access the primary entity data. This saves on the extra call(s) which helps with performance.
- Consider Asynchronous plug-ins over Synchronous: Wherever possible use asynchronous plug-ins over synchronous plug-ins to maintain performance.
- Limit Operations that Cascade to Related Entities: When using the Update method or UpdateRequest message, do not set the OwnerId attribute on a record unless the owner has actually changed. When setting up this attribute, the changes often cascade to related entities, which increases the time that is required for the update operation.

13.7 *Auditing*

13.7.1 Background

- Auditing is the process of tracking changes in Dynamics for security and analysis.
- Auditing is supported for all custom entities and most customizable entities and attributes.
- Below are some of the operations supported by Auditing:
    - Audit of record create, update, and delete operations
    - Audit of relationships (1:N, N:N)
    - Audit of audit events
    - Audit of user access

13.7.2 Auditing Recommendations

- Turn off auditing when doing the initial import of data
    - Disable auditing during the initial load will prevent unnecessary audit records being created helping with reducing wasted disk space
- Do not audit read-only entities and fields
    - Example: Microsoft Dynamics CRM invoices imported from the Enterprise Resource Planning (ERP) system should be read-only
- Choose carefully what is audited
    - Examine the fields that could have a business impact, and enable auditing only for those fields
    - Do not enable auditing on everything and then remove what is not needed. Instead only enable auditing on what is required.
- Delete old log entries
    - Determine the time frame to keep audit logs and incorporate a process to delete the older audit log if business permits

13.8 *Design Constraints and Considerations*

The GCcase platform has a number of deliberate constraints it imposes to prevent any one action having too detrimental an impact on the rest of the system and, therefore, on users.

While this behavior can be frustrating since it can block specific requests from completing and often leads to questions around whether the constraints can be lifted, this is rarely a good approach when considering the broader implications.

When the platform is used as intended and an implementation is optimized, it’s very rare that there is a scenario where these constraints would be encountered. Running into the constraint is almost always an indication of behaviors that will be tying up resources excessively in the system. This means other requests either from the same user or other users can’t be processed. So while it may be possible to loosen the constraint on the request being blocked,

what that actually means is that the resources it is consuming are tied up for even longer causing bigger impacts on other users.

At the heart of these constraints is the idea that the Dynamics platform is a transactional, multiuser application where quick response to user demand is the priority. It's not intended to be a platform for long running or batch processing. While it is possible to drive a series of short requests to Dynamics, it isn't designed to handle batch processing. Equally, Dynamics isn't designed to handle that iterative processing.

In those scenarios, a separate service can be used to host the long running process, driving shorter transactional requests to Dynamics itself. For example, hosting BizTalk or Microsoft SQL Server Integration Services (SSIS) elsewhere and then driving individual create or update requests to Dynamics is a much better pattern than using a plug-in to loop through thousands of records being created in Dynamics CRM.

Understanding platform constraints allows them to be mitigated in the application design. If errors are encountered, understanding the platform constraints helps determine why they are happening and what can be changed to avoid them.

Plug-In timeouts

- Plug-ins will time out after 2 minutes
- Long running actions shouldn't be performed in plug-ins. Avoiding these plug-ins protects the platform, the sandbox service, and ultimately the user from poor user experience

SQL Timeouts

- Requests to the SQL Server time out at 30 seconds to protect against long running requests
- Timeouts provide protection within a particular organization and its private database
- Also provides protection at a database server level against excessive use of shared resources such as processors/memory

Workflow Limits

- Operates under a Fair Usage policy
- No specific hard limits, but balance resources across organizations
- Where demand is low, an organization can take full advantage of available capacity, but where demand is high, resources and throughput is shared

Maximum concurrent connections

- There is a platform default setting of a maximum connection pool limit of 100; connections from the Web Server connection pool in IIS to the database. Microsoft Dynamics does not change this value
- Have never seen a scenario where this should be increased. If the maximum concurrent connections limit is reach, it is an indication of an error in the system
- With multiple web servers, each with 100 concurrent connections to the database of typical <10ms, this suggests a throughput of >10k DB requests for each web server. This should not be required and would hit other challenges well before that

Execute Multiple

- ExecuteMultiple is designed to assist with collections of messages being sent to Dynamics from an external source
- The processing of large groups of these requests can tie up vital resources in CRM at the expense of end user requests, therefore this is limited to 2 concurrent ExecuteMultiple requests per organization
- The actual limit experienced by a customer will first depend on the number of web servers that are available, but also on how requests are distributed across the web servers by the load balancer. While it may be possible to achieve higher than 2, it is safer to design for 2 when not in control of the infrastructure, such as in the online environment

13.9 *Validation*

Some helpful tools to identify possible issues with an implementation of Dynamics are available, they include:

13.9.1 Microsoft Dynamics Custom Code Validation Tool

The Microsoft Dynamics Custom Code Validation Tool helps identify custom web resources that contain unsupported code. The tool, when run against custom JavaScript libraries and HTML web resources, will detect items that are unsupported. The most common issues that this tool targets are:

- Unsupported CRM client SDK calls
- Unsupported CRM end point usage
- Unsupported Common DOM manipulations

The issues that are flagged are unsupported coding processes. Download this tool and extract the contents. The extracted contents include instructions to install and use the tool.

13.9.2 Microsoft Dynamics Server Best Practices Analyzer

To confirm that the system is properly configured, use the Microsoft Dynamics Server Best Practices Analyzer on the server. The CRM Best Practices Analyzer module will test the server and deployment for common mistakes and make optimization recommendations to improve the stability and performance of the deployment.

13.10 *Ux Design Guidelines*

This section describes some of the key guidelines which can be useful while designing the user interface in a solution.

13.10.1 Avoid “drop in” design

It is important to recognize that the changes in the user experience are designed to drive a much more intuitive flow through the application, and a navigation-heavy approach isn’t what many users prefer.

Recognizing and adjusting the design of an application to use navigation more naturally through the dashboards, forms, and views themselves and only using the navigation menu for context switches gives a much more natural experience for the user.

13.10.2 Client-Side Scripting

The predominant cause for poor user adoption is poor performance. One of the major reasons for poor performance is poor scripting habits. Follow these guidelines to help ensure optimal user experience:

• Avoid scripting:

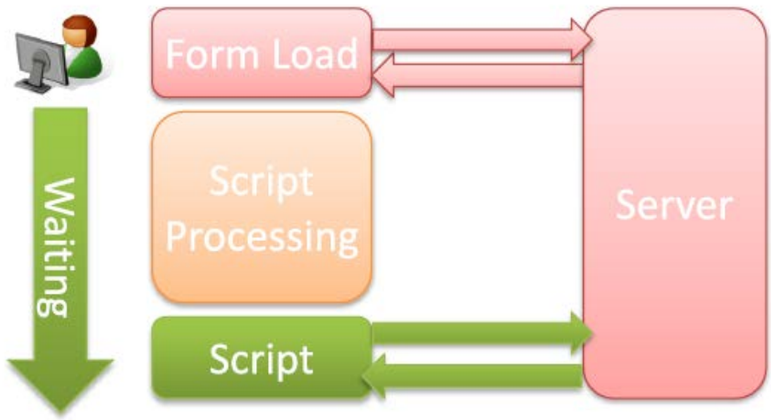
Try to avoid complex client-side scripting. Look at alternative options like server-side code or design change. For example, instead of showing and hiding some fields based on two roles, it might be better to create two roles and use two role-based forms.

• Onload event:

With client-side code, avoid functions with excessive wait times in the onload event. These delays directly add to the form load time. The higher the form load time, the lower the user's perceived application performance.

• Server callback functions:

Server callback functions in client-side code cause the most delay. If a synchronous call is made and the client is waiting for a response from the server – to the user the client looks like it has stopped responding. Although the client is just waiting, the perception the user gets is that CRM is prone to hanging randomly



13.10.3 Custom Theming

For GOC accessibility and look and feel – please see the look and feel section.

The primary purpose behind introducing theming is to help brand the CRM system along the same lines of other corporate line-of-business applications a customer might be running. This does have a tangible effect on user ownership and adoption of the system as well as enhancing the experience when the system is deployed in a customer-facing scenario.

Currently, theming allows for branding the application with a customer icon and changing accent colors for hover and selection of certain areas and entities. Theming isn't meant to enable changes to any of the current layouts or actions.

Dynamics includes a default theme. Keep the following best practices in mind while using the theming functionality:

- Accessibility:
  Be aware of the colour contrast for new custom themes. Contrast ratio is an important measure of accessibility. The out-of-box (OOB) theme has the correct contrast ratios to ensure optimal usability. High contrast mode always uses the Dynamics default colour settings

- Don't overuse colours:
  Although every entity can be a different colour, using one of two patterns is recommended:

  - Make most entities a neutral colour and highlight only the key ones
  - Make entities the same colour when the entities have a similar purpose
  - Keep the number of colour groups low. If too many colours are used, the colours lose their significance and don't add to the user experience.

13.11 *JavaScript*

- Dynamics CRM has multiple extensibility points that allows the application to be extended to meet business requirements
- JavaScript helps with adding client side extensions for users providing rich & responsive experience
- In regards to the Dynamics CRM application JavaScript are added as a WebResource (static, client side content accessible via URL)
- Areas where JavaScript can be used:
  - Form scripts
  - Command bar (ribbon) commands
  - Web Resources (HTML)

When leveraging client side JavaScript within a solution, do NOT reference or manipulate the native Document Object Model (DOM), as this is unsupported. To enable the development of scripts Dynamics provides the "Xrm.Page" model which exposes cross-browser compatible functions for use on a form.

The article in the MICROSOFT DYNAMICS SDK (see referenced documents above) titled "Client-side programming reference" describes the supported functions, methods and techniques that should be leveraged during an implementation.

Do not utilize third party JavaScript libraries such as jQuery in CRM application pages. These libraries introduce an abstraction layer over the DOM. The inner workings of the library may utilize unsupported DOM manipulations. JQuery is however supported when creating HTML and JavaScript web resources.

Use Asynchronous data access methods where possible: for any kind of data operations the browser needs to communicate to the server. Requests that are synchronous in nature will cause the operation to suspend until the response is received back from the server.

Minify JavaScript: Minification is the practice of removing unnecessary characters/white space from the code to reduce the size. In the case of JavaScript, this improves response time performance because the size of the downloaded file is reduced.

Prefer REST endpoint for CRUD operations: There are two web services available to Jscript and Silverlight developers building solutions inside CRM. Both rely on authentication provided by Microsoft Dynamics CRM and can perform CRUD operations while SOAP can perform operations beyond CRUD. The recommendation is to use REST for CRUD operations as it is faster and has a better developer experience. In CRM 2016/D365 there is a new web API based on Odata 4.0 which provides functional parity with the SOAP endpoint

Do not access DOM/Avoid DOM manipulations. Developers can interact with DOM in their HTML web resources though.

Microsoft Dynamics uses many JavaScript objects within pages. A JavaScript developer can discover these objects by debugging a page, and then accessing and reusing these objects. Microsoft reserves the right to make changes to these objects, including removing them and/or changing method names. Thus making direct references to these objects is unsupported.

Use JScript judiciously. While a few lines of JScript can greatly enhance the user experience by automatically updating data as a form loads or as data changes, they can also cause a performance issue if a lot of JScript code is running on a form. When using JScript on forms, verify that code is optimized to ensure that forms are loaded efficiently.

To determine if JScript is affecting the performance of a form, remove the JScript and test how the form performs. Then, compare those results to the performance of the form with JScript included.

Do not reuse any of the available Microsoft Dynamics installed JavaScript code. This code may change or be overwritten during an upgrade.

13.12 *iFrames*

Use iFrames judiciously. The greater the number of iFrames on a form, the greater the associated form load time will be. If an iFrame is not on the form’s primary tab, use collapsed tabs and set the URL in the TabStateChange Jscript event for the tab on which the iFrame resides. This ensures that the iFrame loads only when a user clicks the tab, rather than loading each time a record is opened.

13.13 *Sub-Grid*

Use sub-grids judiciously. Each sub-grid that is used in a form queries the Microsoft Dynamics system in the background for a set of data to load into the grid. While the query executes in the background, each sub-grid control on a form adds more HTML to the page, resulting in longer load times. For sub-grids that are in use, consider having users collapse them, as data will not be retrieved until the user expands a sub-grid. Optimize views and sub-grids so that only the required columns and grids display by default.

Each sub-grid also includes a context-sensitive ribbon to provide users with access to the appropriate commands for that sub-grid. Although the ribbon only becomes visible when the grid is selected, the required xml (and therefore the processing) occurs as the form is loaded. As a result, even if a user never selects the sub-grid, there is a performance impact associated with each sub-grid on a form.

13.13.1 Subgrids vs. Associated grids

Subgrids and lookups are used to provide a natural flow for quick in-context information discovery and consumption. Use subgrids on a form, when the related entity information in question is needed frequently and the number of records that need to be visible for effective consumption is low (<10). It's important to ensure that information that's key to the decision making process, like owner, is visible on the subgrids.

For example, if there are, on average, three to five opportunities per account and the key fields needed in the context of an account are only Opportunity Name, Owner, Estimate Revenue, and Estimated Close Date, it's ideal to show the opportunities associated with an account as a subgrid on the opportunity form. If it was necessary to know what the status of all the opportunities were to determine if they're relevant to the current process, it would be important to add that to the subgrid as well. This avoids the frustration of users needing to open up each opportunity record individually to check the status; a simple change that has huge user experience benefits.

Associated grids are used to provide a more detailed immersive experience for related entity information. Use associated grids on a form when the related entity information in question is needed infrequently or if the number of records that need to be visible for effective consumption is moderate to high (>10). Associated grids take more effort and clicks to get to. However, they provide a dedicated experience to explore and consume a large number of records.

An ideal example is the out-of-the-box (OOB) associated grid for activities on the account form. A large account might have hundreds of activities regarding it. However, it is of interest only in a few situations like escalations or while analyzing the loss of a deal. Hence it makes perfect sense to show the activities associated with an account as an associated grid.

13.14 *Audit*

Every organization's need for its IT systems data audit is tailored keeping in perspective its external and internal regulatory requirements spanning legal, financial and business domains.

While designing and configuring auditing in MS Dynamics, ensure that only the fields that *absolutely need to be audited*, are configured for auditing.

The GCcase program only provides a limited space for storage, costs could be accrued if this limit is surpassed.

Keep in mind that CRM platform performance degrades with the increase of audit data.

Here are some recommendations to reduce the impact of auditing on the performance of the system:

- Turn off entity auditing when doing the initial import of data

- Avoid using CRM auditing on data elements that are received automatically from external systems. In case this is mandatory, assess the automated retry procedures to avoid flooding the audit
  For example:
  - Do not restart a batch from the first element, restart from the element that failed
  - Do not try to restart constantly
  - Try to restart a couple of times and if it still fails, wait 5 minutes before make the next try
- Consider using a custom entity to store the received information as a single record, rather than using CRM auditing to store the information by data fields.
- For read-only data elements and for data that is received from external systems, use a *custom entity* to record the audit information rather than CRM auditing

13.15*Concurrency*

Concurrency is the ability of a database to allow multiple users to affect multiple transactions. In a multiuser system, it often happens that more than one user selects the same record. These users try to save the record with different updates at the same time, and in this scenario, one of the users will have a data loss. To avoid this, enable *optimistic concurrency* when saving and updating the records. The following diagram depicts both the cases where the two users *read* and try to *update* the same record at the same time:

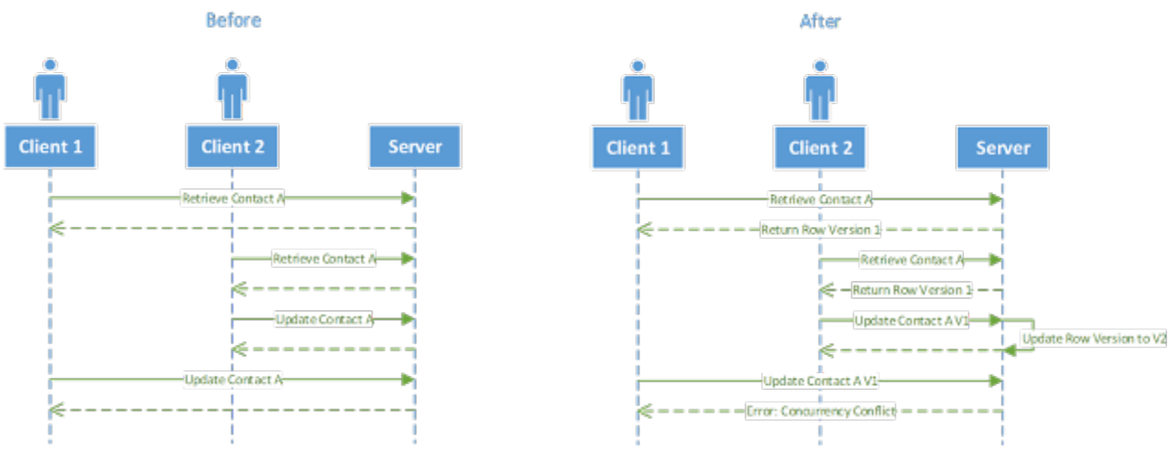


Fig: 1.0 [*reproduced from*] http://community.dynamics.com/crm

The main difference is the ability to check the version of the transaction and update only if the version number matches. Optimistic concurrency is supported on all out-of-box entities enabled for offline sync and all custom entities. For OOB entities, check if the attribute **IsOptimisticConcurrencyEnabled** is set to **true**. For custom entities, this property is set to **true** by default.

13.16*Security*

Modelling

The first step in modeling security is to fully understand the business needs and scenarios that the Solution must support; environments with varying user types and usage patterns have different needs. Attempting to apply one approach or capability to modeling security uniformly across all the different usage patterns often leads to a belief that the only solution that is capable of meeting all the needs is granular, individual access. This is quickly followed by the realization that trying to apply that granular access at higher volumes, for example to managers, becomes a scalability challenge.

In reality, the more common scenario is that a combination of requirements and approaches can be effectively used to meet the needs of both of those groups of usage types, but in a complimentary and overlapping way rather than attempting to use a single uniform model for all.

Consider the following while modelling security:

- Separating and optimizing different usage patterns
- Customizing the security model for different business areas
- Customizing the security model to account for exceptions
- Separating historical data and active data
- Modelling security walls rather than the organizational hierarchy
- Providing separate reporting
- Controlling versus filtering
- Modelling data along security lines
- Security role versus privilege

Security Access

- Double check and store encryption key
- Ensure end users have access
- Ensure end users have AD accounts
- Ensure AD accounts sync'd
- Familiarity with procedure for submitting AD sync issues
- Security review completed

13.17 *Form Design & Views*

Microsoft Dynamics offers a very high degree of customizable development platform. Due to the level of flexibility and extensibility it offers for solution development, there is a potential for performance issues. Performance can be improved by removing unnecessary form fields and optimizing subgrids. This can also result in a cleaner and crisper user experience.

There will always be some minimum load time for forms, however, by optimizing the presented information, the user's experience can be enhanced.

Here are some Dos and Don'ts which can be considered during Form development:

- Avoid including unnecessary JavaScript web resource libraries
  The more scripts added to the form, the more time it will take to download them. Usually scripts are cached in a browser after they are loaded the first time, but the performance the first time a form is viewed often creates a significant impression.

  For a specific example, don't include jQuery in form scripts instead of XMLHttpRequests. While jQuery has the $.ajax function that many people are familiar with to perform these requests, it is a developer preference, not a necessity. It is possible to perform these requests using the native XMLHttpRequest object found in all browsers supported by Microsoft Dynamics CRM. The use of jQuery is not supported. (See key directives)

- Avoid loading all scripts in the Onload event
  If the code only supports OnChange events for fields or the OnSave event, make sure to set the script library with the event handler for those events instead of the OnLoad event. This way loading those libraries can be deferred and increase performance when the form loads.

  It is not recommend to use the addOnChange method within the OnLoad event handler simply as a matter of convenience. While this may reduce the number of steps necessary to add event handlers, it causes the form to load more slowly.

- Use collapsed tabs to defer loading web resources
  When web resources or IFRAMES are included in sections inside a collapsed tab, they will not be loaded if the tab is collapsed. They will be loaded when the tab is expanded. When the tab state changes, the TabStateChange event occurs. Any code that is required to support web resources or IFRAMEs within collapsed tabs can use event handlers for the TabStateChange event and reduce code that might otherwise have to occur in the OnLoad event.

- Set default visibility options
  Avoid using form scripts in the OnLoad event that hide form elements. Instead, set the default visibility options for form elements that might be hidden to not be visible by default when the form loads. Then use scripts in the OnLoad event to display the desired form elements.

- Customize forms to display only the fields required for specific types of records

- Limit the number of rows that are returned per page while still meeting any related business requirements

- After users are sufficiently familiar with Microsoft Dynamics CRM, hide the Get Started pane to reduce any impact it might have on client performance

13.17.1 Quick Create Form

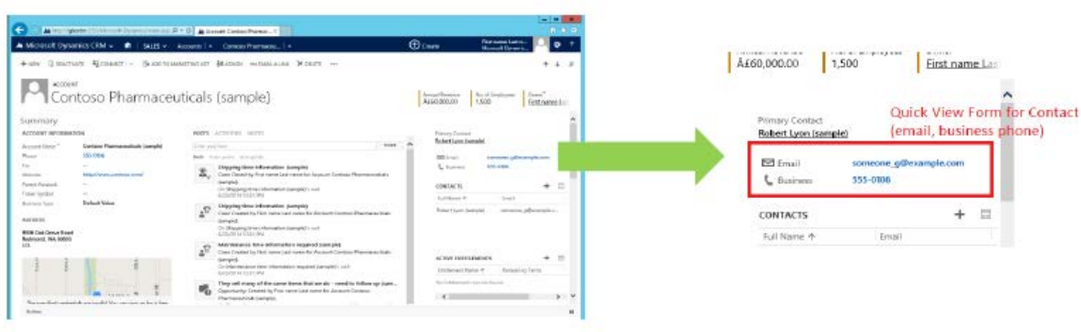
The premise for designing the quick create form was the recognition of the fact that creating content is a different process than consuming content. The intent behind introducing the quick create form was to optimize the experience for creation. As the name indicates, this component was designed for creating content quickly. Hence it is important to only have minimum fields that would always be readily available at creation. The user can transition to the full form for further updates or edits if needed. Quick create forms also allow the main form to be designed and optimized for the consumption experience.

13.17.2 Quick View Form

Quick view forms were created to provide a quick preview of key attributes of a related entity without unnecessary navigation to the related record. This works by surfacing key information on a parent entity form through configuration. They weren't intended for building complete de-normalized forms where multiple fields from multiple related entities are all shown in the parent form. The preceding design option is possible, but it isn't recommended.

Quick view should be used sparingly:

• To show key information only, not replicate the entire record.
• To show information directly actionable or related to the outcome driven from the parent record.



For example, showing the email and phone number of a primary contact on an account form is an ideal use case for quick view forms where the likely action is to contact the primary contact for the account. Rather than exploring the primary contact's full details on the contact form, quick views make it possible to directly contact them without that additional navigation. A contact's full details can be found by navigating to the full contact form.
Note that it's possible to have a different quick view for the same entity with a different parent entity form. The more tailored the quick view forms are to the use case scenario, the more effective it will be. For example, the contact information needed in a quick view form in the context of an open invoice might be different from the information needed in the context of an account (as shown in the illustration). In an open invoice form, it might be important to show the credit limit or credit worthiness of the contact.

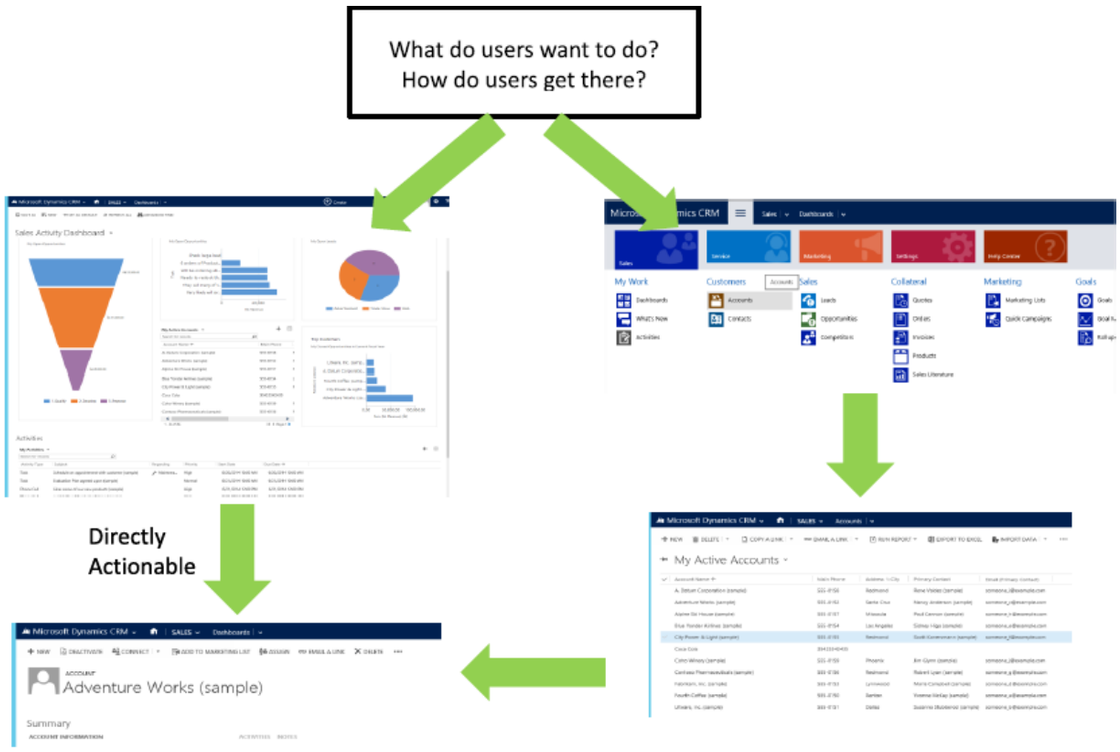
13.18 *Dashboards*

Consider the complexity and number of visualizations used in the dashboards on a page. Dashboards capture a lot of data and are loaded separately in parallel connections to the rest of the page. As a result, limit the number of visualizations used in a dashboard as much as possible without compromising business goals.

13.18.1 Dashboards vs. Navigation bar

Dashboards are a key component to the refreshed user experience. Role-based dashboards ideally should be designed for each role, answering key questions. These dashboards should be the home page from where 80% to 90% of day-to-day tasks for most roles should start. Ideally, all or most business flows should start from the dashboard. This was the key intended use for dashboards by design: to act as that initial gateway into a user's primary work. Consider specific roles individually and provide the information and insight vital to their role, rather than have a generic dashboard for all. The following table can help determine what information to show on a role-based dashboard.

Question	Visualization
What do I need to do to understand the state of business and my work?	Chart of important trends
What are the top tasks for the day?	Queue of activity I need to pick up, My Activities and tasks
What partners/clients do I care about?	View of key accounts and contacts
How am I doing?	Performance



In contrast to dashboards, the navigation bar is used primarily to context switch. This is often misunderstood. The navigation bar isn't intended to be a part of a contextual business flow. It takes multiple clicks and visual searching to find the right area or entity to get to. These tasks make sense for data set exploration, for example, if the user is looking for a set of records via a view for an entity less commonly used. Note this is often an action a developer or customizer would want to do, highlighting the point earlier that the primary solution design should be optimized for the end user rather than necessarily for implementation tasks. However, this sort of task doesn't make sense to be part of a business process as it can cause friction and interruption in the user's interaction with the system. There are better ways to achieve the same effect when the data is needed in the context of their process.

13.19 *Composite fields*

Part of the focus for the optimization of create and consume experiences was recognizing the different ways that structured data is edited as opposed to viewed.

Two particular types of data have been focused on in Dynamics that highlight this principle:

- Names
- Addresses

Take the example of names. Users are used to consuming names in contiguous text, for example, such as the name Henry Wortham.

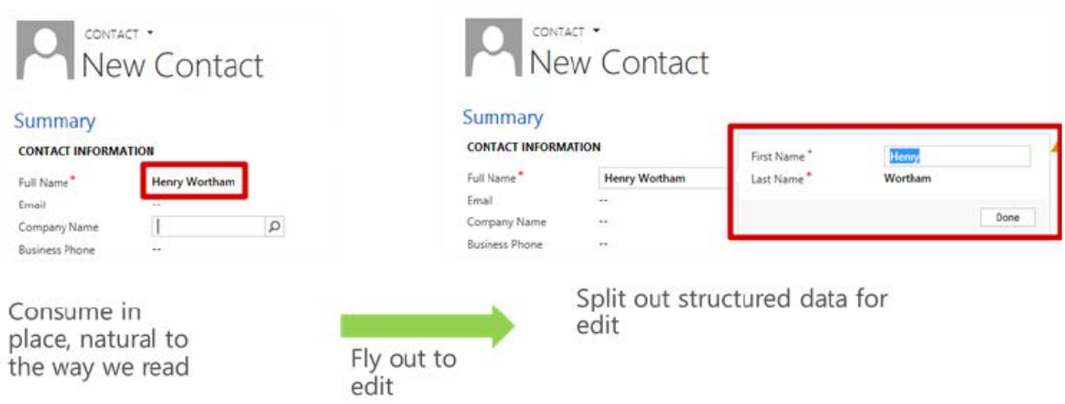
Entering or editing name data however is typically necessary to do differently so that the correct structure is maintained such as being clear which is the first or last name, and whether there is a prefix or suffix.

Entering of a name is best performed therefore in structured form such as:

- First Name: Henry
- Last Name: Wortham

This avoids any ambiguity over which value reflects which structured value, but is not a particular easy form to read and consume.

To this end, on create forms the structured form of capture is always used for these field types. For full forms however, in most cases, the data would be consumed and, only occasionally, updated. Composite fields were introduced to assist with this hybrid experience, offering a more readable form most of the time, but offering through a “fly out” a structured edit experience when the data is to be updated.



With this approach, the Solution is able to provide:

- Optimized experiences through structured capture on quick create forms.
- Optimized consumption experiences on forms when data is to be viewed through static forms of the composite fields.
- Optimized update experience on forms when data is to be edited by offering the fly-out composite field experience to offer structured editing.

13.20 *Process bar*

The process bar was designed to provide an easy way for users to follow an established process. It’s meant to encourage the right outcomes. It is not a wizard for data capture. It is recommended to use the process bar like a checklist to encourage the right practices and to enforce checkpoints at various stages. Therefore, it’s recommended that the number of stages, and the steps per stage, are kept within numbers that are manageable and usable in the UI. This is especially important for mobile and tablet usage scenarios, where long or dense processes will more adversely affect usability.

Another key purpose of the process bar was to link together related entities that are tied to a process through automatic form transitions. This goes hand in hand with the inline navigation paradigm. It is most effective to design a user flow to leverage these UI transitions enabled by the process bar along with other natural navigation methods like lookups and quick views.

For example, the first step to update an existing case might be to associate a contact to the case and create one if needed via the quick create form. In this way, all components of navigation are user-focused for the user flow.

13.21 *Navigational flow*

One of the key principles of the Dynamics user experience is that of an intuitive flow through the application. The intention is that a user should be able to naturally follow a business process, with the system assisting them in achieving their goal, rather than the user feeling like they need to manipulate the software to achieve their ends.

Many of the enhancements have focused on this, in particular:

- In-place navigation rather than pop-up windows:
  - Previously, the use of pop-up windows meant that users lost context of their work as they opened and moved between windows
  - In-place navigation allows the user to more intuitively flow forwards and backwards within a process without having to think about manipulating windows to get back to the information they require

- Auto save:
  - Eliminates the need for the user to need to think or be prompted to explicitly save information as they progress through the application

- Reduced scrolling and clicks:
  - Changes to the overall presentation and flow of data reduces the need to scroll within long forms
  - This was a major cause of additional clicks to access information in previous versions of Dynamics CRM

As will be shown in the next sections, a number of other features were added to assist with the presentation of data in context and the capture of information

13.22 *Application Files*

Directly changing files in the application

Do not change the web application text files. Dynamics is a web application that runs on IIS. Do not navigate to the deployed application folder and make changes to files. All changes are unsupported, including but not limited to: CSS, INI, config, JavaScript, images, html pages etc. Changes to these files are not supported. Rollups and upgrades will overwrite these changes. Access to these files should be restricted. The common platform will not support any changes to these files; the current installation is on premise any customizations will be lost when the Solution is migrated to the common base.

Modifications to the Microsoft Dynamics website (file and website settings) include modifications to the file system access control lists (ACLs) of any files on the Microsoft Dynamics server.

13.23 *Database*

Modifications of the physical schema of the CRM database such as adding or modifying tables, adding functions, stored procedures, triggers or anything else outside of a non-clustered index *is unsupported*.

Note that any indexes added to the database should be managed in such a way, that they are checked and re-applied if removed by any rollup up or patch.

13.23.1 Retrieving Data Directly from Database Tables

The database should be treated as a black box. Accessing the tables directly by utilizing a database level account bypasses the security infrastructure and *is not supported*. Furthermore, the table structure can change during upgrades and rollups.

13.23.2 Updating Data Directly in the Database Tables

Developers should only use the API provided to update data. With Microsoft Dynamics, on-premises do not access the CRM data directly in the database tables. *Accessing data in this manner is unsupported and invalidates vendor support*. Furthermore rollups and application updates will overwrite any changes. Developers should always use the APIs provided with the application platform web services to update data.

13.23.3 Changing the Database Tables, Stored Procedures, or Views

Utilizing database tools to change the database is unsupported with the exception of adding or updating indexes. Customization tools should be used to add any new entities or entity attributes. This is the only supported way to apply changes to these parts of the database. Any direct changes made risk breaking the application or the ability to apply update rollups. Any changes applied may be destroyed when an update is applied, or during an upgrade, and any data that may have been included in custom database table columns will be lost.

13.24 *Plug-ins*

GCcase supports the execution of plug-ins and custom workflow activities in an isolated environment. In this isolated environment, also known as a *sandbox*, a plug-in or custom activity can make use of the full power of the Microsoft Dynamics SDK to access the organization web service. Access to the file system, system event log, certain network protocols, registry, and more is prevented in the sandbox. However, sandbox plug-ins and custom activities do have access to external endpoints like the Microsoft Azure cloud service. Access to the underlying file system is unsupported thus the use of a custom entity for tracing is suggested.

13.25 *Quick Find*

Customizing quick find views by limiting search columns

When using the quick find feature for an entity, the results are displayed in the entity's Quick Find view. When customizing this view, fields can be defined for the columns returned, sorting and filter criteria, and the columns (fields) that will be searched.

The number of fields that are searched to display Quick Find results can directly affect performance. To ensure optimal performance, configure the Quick Find feature for an entity to search only the fields necessary to address specific business requirements.

To customize Quick Find search parameters:

1. On the Microsoft Dynamics home page, in the navigation bar, click Settings, under Customization, click Customizations, and then click Customize the System.
2. In the Solution window, under Components, expand Entities, expand the desired entities to customize the Quick Find view for, and then click Views. In the list of available views, select the Quick Find view, on the action toolbar click More Actions, and then click Edit.
3. In the View: Quick Find window, under Common Tasks, click Add Find Columns.
4. In the Add Find Columns dialog box, specify the fields that will be searched to provide Quick Find results, and then click OK.
5. In the View: Quick Find window, in the action bar, click Save and Close, and then in the Solution window, click Publish All Customizations.

When customizing Quick Find search parameters consider the following points:

- The more columns included in a search, the longer the search will take
- Fields included in a search should be leading columns in indexes, even if this means creating one for each field. Also consider including in those indexes the Owner, BU, and the State fields, which are typically included in the query
- Using filtered indexes can result in better query performance
- Limit the number of columns selected only to those necessary to meet business requirements. Quick Find queries that include more than eight-to-ten well-suited columns (as defined below) may adversely impact performance
- Quick Find functionality is optimized for locating one record or a small set of records. As a result, it is recommended to include search fields that mostly include unique values across records. Searching on phone numbers, email addresses, names, or other unique values (including custom attributes that are likewise well suited for Quick Find searches) will yield satisfactory results. On the other hand, searching on address data (e.g. postal codes, city name, and so on), while supported, is discouraged because values in these field tend to be common across multiple records
- Avoid searching on pick list attributes; this would result in a search of label that can be localized, and these types of columns currently cannot be indexed

- Avoid search terms that begin with a wildcard character (e.g. using an asterisk (*) as the first character) – queries using these types of search terms cannot use indexes to optimize performance
- Avoid using search terms that are not very selective

These processes are described in this Optimizing and maintaining the performance of a Microsoft Dynamics CRM server infrastructure

13.26 *Reports*

The solution provides out of the box reporting capabilities, but in complex reporting scenarios some clients will require more capabilities. Please contact the GCcase Program Centre to discuss reporting requirements and performance options.

Please note that GCcase may not support third party reporting extensions / plug-ins.

13.26.1 RDL Sandboxing

The RDL (Report Definition Language) Sandboxing feature lets the GCcase program detect and restrict the usage of specific types of resources, by individual tenants, in an environment of multiple tenants that use a single web farm of report servers.

When RDL Sandboxing is enabled, the following features are disabled:

- Custom code in the **<Code>** element of a report definition
- RDL backward compatibility mode for SQL Server 2005 Reporting Services (SSRS) custom report items
- Named parameters in expressions

13.27 *AD Sync*

The Microsoft Dynamics Active Directory user sync solution was originally designed to fulfill the customer's common requirement to keep user data in sync with the source directory, a feature not offered by the out-of-box dynamics product. Here are the current features that version 1.0.3 offers:

- Automatically maintain up-to-date system user data that's consistent with windows' active directory services.

  Here are the attributes that gets updated if available:

title	**Job Title**	**string**
firstname	First Name	string
middlename	Middle Name	string
lastname	Last Name	string
internalemailaddress	Primary Email	string
address1_telephone1	Work Phone	string
address1_telephone3	Pager	string
address1_fax	Fax	string

homephone	Home Phone	string
mobilephone	Mobile Phone	string
address1_postofficebox	Post Office Box	string
address1_line1	Street 1	string
address1_city	City	string
address1_postalcode	ZIP/Postal Code	string
address1_stateorprovince	State/Province	string
address1_country	Country/Region	string

- Allow for flexibility in changing preferred username and ensure system users are identified by their most current username in Dynamics CRM
- Minimize Dynamics license exposure for users tied to disabled/removed domain accounts

13.28 *Security Roles*

Delegate configuration tasks to users with the System Customizer security role and test any configurations made with a user account that doesn't have customization privileges. To accomplish this effectively, utilize two user accounts: One account with the System Customizer security role and another that has the security roles that represent the people who will be using the customizations.

The best practice analyzer gives specific guidance on security roles and permissions.

Guidelines for security roles:

- The number of security roles must not exceed 250 roles per tenant
- No single user will be part of more than 500 teams
- Never use the Out-Of-Box Security roles, rather always clone them
- Security roles are **additive,** so when 2 roles are in conflict on a permission, the greater permission is given. In other systems, this is sometimes not the case
- Business Units are data security related and not a business hierarchy
- An organization must not create more than 250 Business Units
- Security Roles are roles not job titles
- Limit sharing to a minimum. If sharing is necessary, share to teams
- Do not let anyone function as system administrator or system customizer roles. IT and administrators should log in with special accounts to make changes. Example CRM Admin account (Which would have System Administrative access)
- Keep the number of security roles as minimal as is practical
- Use meaningful role names
- Roles should be copied, renamed and modified; the default roles should remain unchanged

13.29 *User Data Access*

The more teams a user belongs to, the greater the potential that the user will experience a performance impact. Balance the need to limit the number of teams that a user belongs to with the need to develop a manageable business unit structure.

The more records that are shared with a user, the greater the potential that the user will experience a performance impact. Be judicious with the amount of sharing that occurs within a specific implementation – share as few records as possible while still accomplishing overall business goals.

For more information: Optimizing and maintaining the performance of a Microsoft Dynamics CRM server infrastructure

13.30 *Application Programming Interface (API)*

The use of application programming interfaces (APIs) other than the documented APIs in the Web services CrmDeploymentService, CrmDiscoveryService, CrmService and MetadataService is not supported. For more information see the following: Supported extensions for Microsoft Dynamics (D365):

13.31 *Outlook*

Modifications to any one of the Microsoft Dynamics forms or adding new forms such as custom .aspx pages directly in Microsoft Outlook or changes to .pst files. These changes will not be supported.

13.32 *Web*

Do not reuse any Microsoft Dynamics User Interface controls, including the grid controls. These controls may change or be overwritten during an upgrade. Changes in the default.css file in the Microsoft Dynamics root installation folder are not supported.

Do not reuse any available Microsoft Dynamics installed JavaScript code. This code may change or be overwritten during an upgrade

Creating an IIS application inside the Microsoft Dynamics Website for any Virtual Directory and specifically within the ISV folder is not supported.

13.33 *Configuration Tools*

Perform configuration to the application using the application tools. All configurations performed using the tools are supported by Microsoft. When Microsoft Dynamics is configured using the configuration tools in the application, metadata is modified, the core structure remains unchanged. Rollups and upgrades take this data into account and new features can be added to CRM without impacting custom configurations.

If further enhancements are required, CRM can be extended. Developers who extend Microsoft Dynamics have a responsibility to follow rules and best practices documented in the Software

Development Kit for Microsoft Dynamics (D365). The SDK documents the APIs available to developers and provides guidance about how to best use them. Microsoft supports the APIs and practices that are documented in the SDK. APIs and programming practices not documented in the SDK are not supported by Microsoft. APIs and code on the internet may describe how a problem can be solved, but if it doesn't leverage APIs documented in the SDK, it is not supported. Before a developer applies a change, it should be verified whether it uses supported methods.

13.34 *Development Practices*

Configuration should always occur within the context of a solution. The Program Centre will assist in setting up a best practice methodology for managing the organization`s configurations. A common development practice for CRM is to develop locally on VMs because it's safe to have access to the database/application levels in a closed environment and much easier to troubleshoot.

13.35 *Look and Feel*

Re-skinning of the core application is not supported. To achieve WCAG 2.0 compliance for a web front end, utilize the CRM API. API calls can be performed via HTTP requests from a WCAG 2.0 compliant web front end.

See custom theming section.

13.35.1 Solution Publisher

Every solution has a Publisher. The default solution has a publisher named "Default Publisher for <organization name>".

The publisher record contains a Prefix value. The default value of this prefix is "new". When a new solution component is created, this prefix will be appended to the name. This is a quick way that allows users to understand the relationships between the Solution and components.

The Program Centre will assign the Prefix value for the default publisher to something that identifies the organization.

14 Performance

14.1 *Exploring Systemic Performance Problems*

When the platform is used as intended and an implementation is optimized, it's very rare that there is a scenario where the constraints, *detailed above,* would be encountered. If performance problems persist please contact GCcase.

In some cases developers / testers may require trace files or other system level logs, please contact the GCcase program office for options. For reference please see the following: Microsoft Dynamics deployment-level tracing

Running into the constraint is almost always an indication of behaviors that will be tying up resources excessively in the system.

No CRM activities, either core platform or implementation, work completely in isolation.

14.1.1 Common Symptoms

These types of problems typically exhibit a combination of common symptoms as given below:

- Slow Requests
  - Users see slow response times for the system in particular areas, for example, certain forms and queries
- Generic SQL Errors
  - Certain actions respond with a platform error reporting a "Generic SQL Error"
  - This often translates at a platform layer to a SQL timeout
- Deadlocks
  - Platform errors reporting that a deadlock has occurred, which has forced the action to be terminated and rolled back
- Limited throughput
  - Particularly in batch load scenarios, this often exhibits in really slow throughput being achieved, much slower than should be possible
- Intermittent Errors / slow performance
  - An important indicator of these behaviours is where the same action can be very fast or incredibly slow, and retrying it works much more quickly or avoids an error

Common symptoms have causes that force particular requests to run slowly and then to trigger platform constraints.
Given below are the typical symptoms with some of the common root causes of these symptoms.

- *Slow Requests* → Complex queries →Blocking → Long running transactions
- *Timeouts* → Complex queries → Blocking → Long running transactions
- *Deadlocks* → Parallel requests for same resources → Blocking → Long running transactions
- *Low Throughput* → Blocking → Long running transactions

The underlying impact of long running transactions, database blocking, and complex queries can all overlap with each other and amplify their effects to cause these symptoms. For example, a series of long running queries that are completely independent of each other may cause slow user response times, but only once they require access to the same resources do the response times become so slow that they become errors.

Given symptoms listed above and the resulting output of long running transactions, one can almost always assume (exceptions apply, of course) that one of the key reasons for long running transactions is '*blocking'* which results from '*locking'* of database resources due to variety of reasons.

So, one could come to the conclusion that locks (or blocks) are the key reason for most of the performance related issues/errors that one encounters in their solutions. This conclusion can be summarized as given below:

Locks/blocks → Slow user response time's → Unpredictable response times → Low throughput in batch processing → Errors.

The initial issue is locks and therefore *blocking* in the system. This leads to slow user response times, which results in unpredictable and unreliable user response times, often in a particular area of the system.

In the extreme case or under heavier than normal load, this can then show through in any background batch processing with poor throughput. Eventually it can all escalate into errors occurring in the system.

14.1.2 Locking and Transactions

Although the details of how this is done is beyond the scope of this document, the most simple element to consider is that as Dynamics interacts with data in its database, SQL Server determines the appropriate locks to be taken by transactions on that data such as:

- When retrieving a particular record, SQL Server takes a read lock on that record
- When retrieving a range of records, in some scenarios it can take a read lock on that range of records or the entire table
- When creating a record, it generates a write lock against that record
- When updating a record, it takes a write lock against the record
- When a lock is taken against a table or record, it's also taken against any corresponding index records

Note: It's possible to influence the scope and duration of these locks. It's also possible to indicate to the SQL Server that *no lock* is required for certain scenarios.

Here are some recommendations on how to approach transaction locking in the Solution:

- Lock only when necessary. Use 'Nolock hint' when required. Optimize locking requirements.
- Consider order of locks
- Hold contentious locks for shortest period
- Reduce length of transactions
- Optimize requests
- Reduce chain of events
- Avoid multiple updates to the same record
- Only update when absolutely necessary
- Multiple triggers on same event
- Examine and decide when to use different types of customization

- Plug-ins/workflows aren't batch processing mechanisms
- Security is a costly operation
- Diagram related actions
- Isolation modes

Please note that details are available at: Microsoft Dynamics 365 Performance and Scalability Documentation

Three key solution developmental aspects (i.e. Database, Design and Platform) can be used to summarize the systemic problems often encountered by solutions during development and after deployment as given below:

- Locks/ transactions
  - Locks and transactions are essential to a healthy system
  - But when used incorrectly, can lead to problems
- Platform constraints
  - Platform constraints often exhibit in the form of errors
  - But rarely is the constraint the cause of the problem
  - They're there to protect the platform and other activity from being affected
- Design for transaction use
  - If implementations are designed with transaction behaviour in mind, this can lead to much greater scalability and improved user performance

14.2 *Guidelines for Solution Performance*

This section aims at giving recommendations for better performance of a CRM Solution.

In particular, it provides a summarized checklist that the *Tenant CRM development team* can use as guidance during configuration, development and testing of their solution and more specifically, the custom components (Plug-Ins).

The eventual goal, of this (*Development & Prototyping Directives)* document, is to ensure that tenant-manufactured software components do not put the stability of the GCcase platform at risk.

Issues, which can be avoided by a sound design and smart testing, should get addressed before the Solution reaches the pre-prod environment.

14.2.1 Performance Checklist

Below are some of the key guidelines which can impact the performance (and hence the stability of the platform) of the Solution. Consider the checklist as the summary of the Solution design and performance considerations given in Section 12 and 13 of this document, and as an evolving list which will grow as more understanding is gained from lessons learned, and as more tenants are introduced on the platform.

An attempt has been made to include the most common performance impacting guidelines, tips and design considerations, related to performance and platform stability.

- Approach to ***building solutions*** - make *logical groupings* as shown below.
    - Using this approach eliminates the requirement to re-publish the entire solution when only forms are to be updated.
    - Solutions:
        - Custom Plug-ins (solution)
        - Custom Entities (solution)
        - Default Entities (own solution)
        - Options Sets (solution)
        - Process Sets (solution)
        - Forms (solution)

- Security – 'keep it simple' should be the motto!
- Avoid deep Business Unit (BU) access levels
- Turn *auto save* option OFF
- Avoid sharing records with users. e.g. use teams
- Use more plugins/Workflows then Business Rules instead of JavaScript removing the load on the browser
- The default value for the number of records shown on the screen is 25.
    - In case there's a need to increase it during work session, please remember to set it back to its default value.

Configuration/Customization

- **Forms**
    - Limit the number of fields
        - There is no metered count, but the fewer fields on the form, the better it performs
    - Role-based forms
        - Should be used to help out with the previous point
    - Collapse Tabs
        - Try and use collapsed tabs where possible to have better load performance
        - Iframe and subgrids should be loaded through collapsed tabs where possible
    - Be mindful when attaching JavaScript libraries to the form

    * Limiting the data to be loaded on the form helps with SQL, and Network & Rendering

- **Views**
    - Limit the number of fields
    - Limit the number of rows returned

    *Both points above will help in reducing the SQL pressure and the network traffic

    - Quick Find Views
        - Keep the find columns to a minimal (3-4). More find columns results in the underlying query becoming more inefficient leading to poor performance
        - Compliment the queries by adding custom indexes for find columns that may not be already indexed
        - Following the practices above there can be significant performance improvement when performing search
- **Dashboards**
    - Limit the data (both at a column level and pre-filtering level)
    - Remove redundant items
        - A lot of times there will be the same dataset that gets put on the dashboard but with slightly different filter, e.g. Active Contacts & Inactive Contacts
        - As a workaround if the dataset is pretty close then provide options for the users to switch the view as needed
    - Keep it visual in form of charts, etc. (as the aggregate information performs better than raw data)

    - Iframes are good to reduce some pressure by loading some pages that are loading separately from CRM server(s), i.e. external application
    - The next 2 items are not directly related to onboarding but may need to be visited in the long term:
        - With the usage there may be some performance issues, that may demand performance tuning the backend queries as needed
        - Along with performance tuning regular SQL maintenance should be done
- **Reports**
    - Review the number of fields returned in each dataset
    - Ensure unused datasets in the report definition are removed, as all the datasets on the report get executed regardless of whether they are used adding unnecessary stress on the system
    - Use No lock hint on the data retrieval queries where applicable, applies to FetchXML as well
    - Add filters and parameters to reduce the data retrieval
    - Use GROUP BY judiciously: try to implement Grouping at the dataset level vs the report as it will help reduce the data transfer and processing on SSRS engine
- **Miscellaneous**
    - Disable IM Presence if not being used
        - Option present under Settings->Administration->System Settings
        - If the option is enabled calls are made to the PresenceService.asmx, and if the option is not being made these would be unnecessary calls that can be removed
    - Enable Quick Find record limits
        - Another system setting which can be found under Settings->Administration->System Settings which is used to control extensive searches
        - There is a deployment property (QuickFindRecordLimit) which comes into play when this setting is enabled. The default value for this 10000 record. If the search returns records more than 10000 an error is thrown
        - Though the deployment property can be modified to a higher number it is recommended to keep it low
    - Disable grid page counts if not that important for the end users
        - It is an organization level setting that is controlled via SkipGettingRecordCountForPaging key using the OrgDBSettings tool
        - Disabling this setting will remove the count query that gets performed on every subgrid that is loaded on the page
    - Workflows
        - Check logic for infinite loops
        - There is execution depth of 8 after which the workflow will be cancelled
        - Running workflows in asynchronous mode is recommended but that adds records of execution in a few tables that when grows beyond a limit will have some performance impact
        - To control this growth there is a setting to “Automatically delete completed workflow jobs” in the workflow definition screen
        - If retaining the completed workflow log is required, please contact the GCcase program office.

Client

- Client Hardware
    - X86/x64 bit 1.9Ghz or faster dual core processor
    - 4 GB recommended
- Power settings for the client machine recommended to set for “High Performance”
    - Especially important if using Dynamics for Outlook & Dynamics for Outlook with Offline Access
- Network Characteristics
    - Latency should be less than 150 ms
    - Bandwidth should be greater than 50 KB/s
    - If the above minimum is not met there could be significant performance impact
- Internet Explorer Settings
    - Increase the browser cache size to ensure the browser is capable of taking advantage of improved load times

    - Internet Options – General – Browsing History – Temporary Internet Files and History Settings – Disk space to use: 250MB
- Prevent the cache from being cleared every time browser is closed
    - Internet Options – General – Browsing History. Verify that “Delete browsing history on exit”: unchecked
- Update the cache automatically
    - Internet Options – General – Browsing History – Temporary Internet Files and History Settings – Check for newer versions of stored pages
- Internet Explorer Zoom Setting
    - Should be set to 100%
    - Anything over can increase the rendering time and affect the performance

14.3 *Monitoring and Analyzing*

In some cases developers / testers may require trace files or other system level logs, please contact the GCcase program office for options. For reference please see the following:

Microsoft Dynamics deployment-level tracing

Monitoring and analyzing system performance and health is the key to any system implementation. In order to make sure that the CRM application is performing to its expectations, the following are some of the key items to take care of during testing:

- Ensure performance metrics are captured during testing
- Ensure concurrency tests are conducted to mimic production use
- Ensure performance metrics are compared against expected metrics

The following sub sections explore more details on these two aspects.

14.3.1 Monitoring

The following are the key aspects of how to monitor the health of the application:

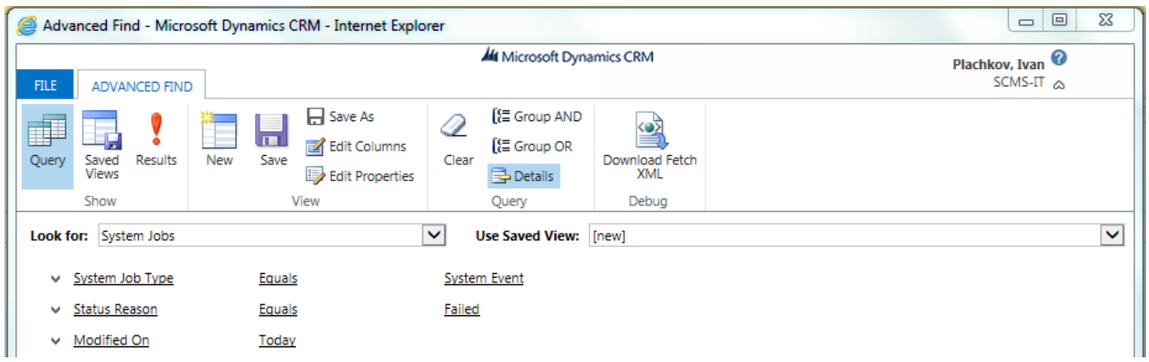
- Monitor Performance with F12 Developer Tools
- Use CRM Performance Center (CTRL+SHIFT+Q) to troubleshoot performance issue related to loading CRM forms.
- Use CRM diagnostics page of the tenant to run CRM Latency and Bandwidth Tests

  https://*<TENANT-URL>*/tools/diagnostics/diag.aspx

- Monitor ‘*exceptions’* generated by the Asynchronous Registered Plugins.

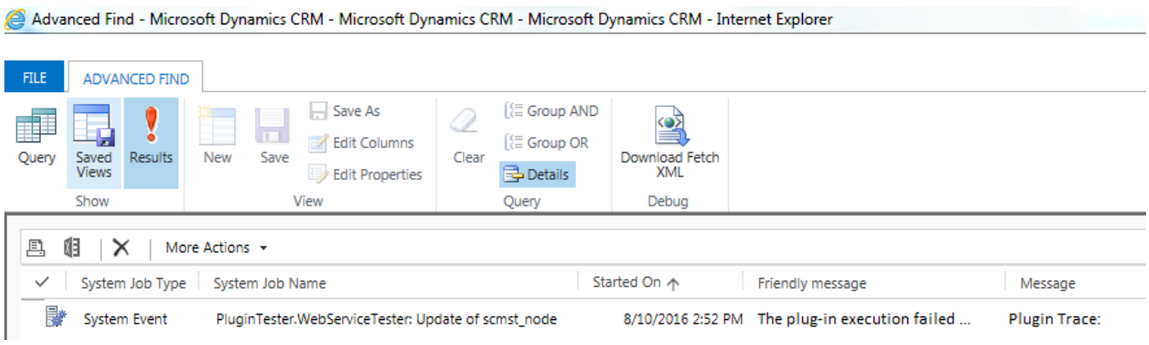
  Microsoft Dynamics uses System Jobs entity to track many features like: Send Bulk E-Mail, Import Data, Workflows, Duplicate Detections, Plugins, etc. These jobs are processed by the Asynchronous Service and it can happen that the service fails to process them because of unhandled exceptions or timeouts. **System jobs with a status reason of failed are mainly from plugins and workflows with logical errors in them.**

  CRM **Advanced Find** provides more information about the Asynchronous Plugins and **their failures reasons**. Please have in mind that Asynchronous Plugins are of type **System Event** in CRM and the *Name of the System Job* is the name of the plugin.



The attribute **Message** may include Plugin Trace with detailed error message for unhandled exceptions, timeouts, etc. Here is one example for a sandboxed asynchronous registered plug-in*: The plug-in execution failed because the operation has timed-out at the Sandbox Client.*

Ensure that **System Job Name, Message Name, Friendly message** and **Message** are included in the results.



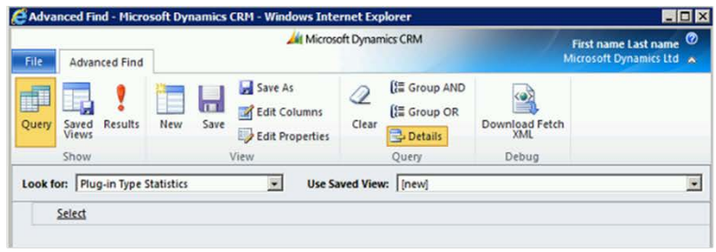
14.3.2 Analyzing Plug-in Statistics

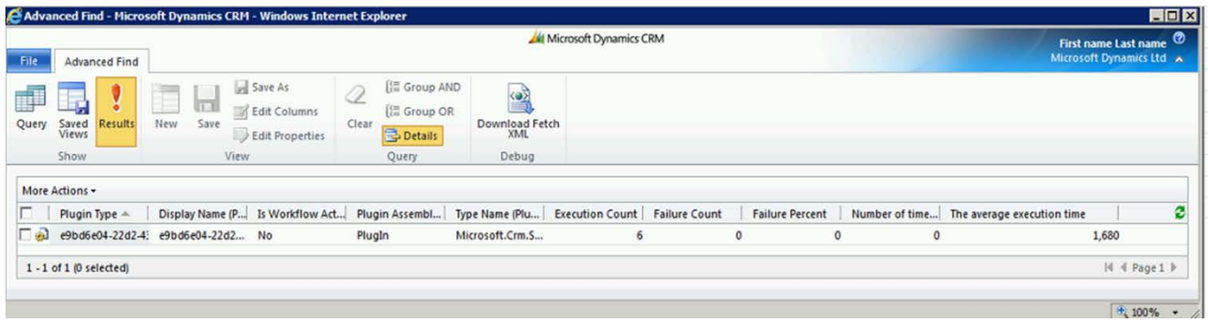
For plug-ins operating in the sandbox, plug-in statistics can be queried for within the application to give an indication of how often the plug-in is running, and statistics on how long it typically takes to run. This information is stored in the database using **PluginTypeStatistic** entity records. These records are populated within 30-60 minutes after the sandboxed custom code executes.

Here are some important PluginTypeStatistic attributes:

Display Name	Description
Plugin Type	Unique identifier of the plug-in type associated with this plug-in type statistic.
Execution Count	Number of times the plug-in type has been executed.
Failure Count.	Number of times the plug-in type has failed.
Failure Percent	Percentage of failures for the plug-in type.
Number of times crashed	Number of times the plug-in type has crashed.
Percentage of crashes	Percentage of crashes for the plug-in type.
Percentage contribution to crashes	The plug-in type percentage contribution to crashes.
The average execution time	The average execution time (in milliseconds) for the plug-in type.
Modified On	Date and time when the plug-in type statistic was last modified.

The name of the Plugin can be retrieved from the attribute Name hosted in entity Plug-in Type.





15 Integration

The Program Centre will create integration points to GC standard enterprise applications with the assistance of the community of users.

The preferred method of integration to Dynamics is via CRM Web Services. The complete SDK and related sample is available here: Software Development Kit for Microsoft Dynamics

The Microsoft Dynamics Software Development Kit (SDK) is for developers, system customizers, and report writers. This SDK documentation contains information for developers writing server side code, custom business logic, integration modules, workflow assemblies, and plug-ins. It provides an architectural overview of Microsoft Dynamics CRM, the entity model, security model, and web services. Sample code and walkthroughs are provided to guide developers through the new features. It also contains information for developers customizing the web client or Microsoft Dynamics for Microsoft Office Outlook, including scripting, integration of custom web pages, and sample code.

In addition to the documentation, this download package includes the assemblies and tools needed for development, helper code for authentication, and Microsoft Visual Studio projects for sample code found in the documentation. Look for regular updates to this SDK.

The User Interface Integration (UII) solution framework can be downloaded, which includes a deployment guide, development guide and API reference. UII uses Microsoft Dynamics for the delivery of configuration data for the Integrated Agent Desktop. It includes development and run-time components. Applications built with UII can provide unified access to customer information across different systems and can aggregate different modes of customer interactions or channels.

Description of utilizing CRM services in code: Use Microsoft Dynamics 365 services in code (D365)

15.1 *Document Management*

The Program Centre will create core integration points to GCDOCS with the assistance of the community of users. A GC Docs integration solution developed by Health Canada is available for download through the code repository, furthermore the functionality of the solution continues to be developed by the GCcase program. Please note that the GCcase Dynamics platform is not a document repository, clients with document storage requirements should plan for a repository, and not store a large volume of documentation in CRM.

15.2 *Outlook*

If using the Outlook plug-in for mail integration, the software will need to be installed on the client workstation.

Reference document: Set up Dynamics for Outlook

Dynamics for Outlook, the legacy add-in for Outlook, is a full Dynamics client that includes offline capabilities. However, as of the December 2016 update for Dynamics (online and on-premises), the preferred way to use Dynamics and Outlook together is to use Microsoft Dynamics App for Outlook paired with server-side synchronization.

With Dynamics App for Outlook (not the same thing as Dynamics for Outlook), Dynamics information appears next to a user's Outlook email messages or appointments. They can view information about contacts and leads stored in Dynamics and add Dynamics contacts directly from an email message. They can also link email, appointment, and contact records to new or existing Dynamics records, such as opportunity, account, or case records. Dynamics App for Outlook is very simple to deploy and it works with Outlook on the web (included in Microsoft Office) the Outlook desktop client, and Outlook mobile: Set up Dynamics for Outlook

15.3 *Exchange*

The CRM services will connect to exchange via server side synchronization over Exchange Web Services (EWS) or a combination of Post Office Protocol (POP) and Simple Mail Transfer Protocol (SMTP). The recommended approach for managing the end user mailbox items – emails, appointments, contacts and tasks – is by using an impersonation role over EWS. Service accounts and shared mailboxes are recommended for email synchronization (over POP/SMTP or EWS) of CRM queues and CRM message notification. Depending on the lines of the business and specific client processes various combinations are possible.

From a **GCcase program centre** point of view, the Email integration requirements should be known ahead of time before configuring a tenant. The administrator should know if Appointments, Contacts and Tasks (ACT) will be synchronizations between GCcase and Exchange Server. This will help the administrator to design the tenant's **Email Profiles** and subsequently associate the End User, Generic Account and Queue **CMR Mailboxes** to them.

- An Email Profile stores settings that are used by server-side synchronization to connect to an Exchange server and process email from the associated mailboxes. At least one email server profile to process email is required.
- Every user account or queue created in CRM has an associated Mailbox record in CRM. Each mailbox can be associated with zero or one profile.

- When the mailbox is not associated to a profile, the user's email activities in GCcase are not synchronized with Exchange server, but there is an option for them to be synchronized through Outlook.
- If the design leads to the use of multiple email server profiles, one of them must be selected as the tenant default profile. It will be assigned automatically to the newly created user accounts.
- For server side synchronizations email addresses must be approved.
- If generic mailboxes/service accounts are required, they have to be created before starting the configuration and the tenant administrator should have access to the passwords.
- Auto discovery option is not available for GCcase environments. The administrator must know the Exchange Server location (URL) and the port.
- For Exchange server located in Legacy or a Partner Datacenters (not ETI), the server certificate and the URL must meet specific requirements:
    - the URL must be listed in the Subject Alternative Name property of the certificate,
    - the certificate validation must be based on Entrust, and
    - the URL must be resolved on GCcase servers to GCNet IP address.

Email Configuration

Configuring email integration.

GCcase Email Configuration - Best Practices: Email Configuration

Exchange Impersonation

Permissions required for exchange integration

Reference document: Exchange Impersonation

15.3.1 Functionally

One of the Dynamics CRM use cases is to view all customer communications in one place, so anyone with the appropriate permissions can see all relevant customer records. For example, view all email associated with a particular contact, account, opportunity, or case.

To store email and other messaging records in Dynamics, synchronize the email system with Dynamics. There are two ways to do this:

- Server-side synchronization
- Microsoft Dynamics for Outlook (includes a synchronization agent). However, see the deprecation notice below.

Server-side synchronization is the preferred synchronization method for the following reasons:

- **Enables Exchange folder tracking**. With folder tracking, users can simply drag email to an Exchange folder to track it automatically in Dynamics. Folder tracking works on any mobile device that supports Microsoft Exchange, which means users can track email from just about any device. Track Outlook email by moving it to a tracked Exchange folder
- **Automatic synchronization**. When synchronizing records with server-side synchronization, the synchronization happens automatically at the server level. This isn't true when synchronizing records with Dynamics for Outlook. In this case, the user has to have Dynamics for Outlook open to synchronize records.

- **Enables multiple scenarios, including hybrid scenarios**. Use server-side synchronization to connect:
  - Dynamics Server (on-premises) to Exchange Server (on-premises)
  - Dynamics Server (on-premises) to Exchange Online
- **Synchronize appointments, contacts, and tasks**. In addition to email, Outlook can synchronize appointments, contacts, and tasks.
- **Synchronize with POP3 email servers**. Use server-side synchronization to synchronize Dynamics with Gmail, Outlook.com, Yahoo, and other POP3 email servers. Note, however, that appointments, contacts, and tasks with POP3 email servers can't be synchronized.
- **Integrated mailbox management and resource utilization**. Use the server-side synchronization performance dashboard to quickly monitor mailbox performance across the organization. Troubleshoot errors through error logging and reporting.

For more details please refer to: Microsoft Exchange Integration Guide

16 Data Migration

Import data into Microsoft Dynamics 365 by using the data import feature. Data import allows the ability to upload data from various customer relationship management systems and data sources into Microsoft Dynamics. Data can be imported into standard and customized attributes of most business and custom entities. Related data, such as notes and attachments can also be included. See: Data Import

16.1 *Bulk Load*

The bulk loading process can be very process intensive. Please contact the GCcase Program Centre to coordinate scheduling for large data imports.