

Microsoft Dynamics CRM

Microsoft Dynamics CRM as a Solution Development Framework

Architecture Whitepaper
Simon Hutson, Microsoft Limited

March 2010



The information contained in this document represents the current view of Microsoft on the issues discussed as of the date of publication and is subject to change at any time without notice to you.

This document and its contents are provided AS IS without warranty of any kind, and should not be interpreted as an offer or commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.

The descriptions of other companies' products in this proposal, if any, are provided only as a convenience to you. Any such references should not be considered an endorsement or support by Microsoft.

Microsoft cannot guarantee their accuracy, and the products may change over time. Also, the descriptions are intended as brief highlights to aid understanding, rather than as thorough coverage. For authoritative descriptions of these products, please consult their respective manufacturers.

©2010 Microsoft Limited. All rights reserved.

Table of Contents

1	Executive Summary	4
2	Microsoft Dynamics CRM as a Solution Development Framework	5
2.1	Introduction	5
2.2	Lower Cost	5
2.3	Faster Development	6
2.4	Leveraged Upgrades	7
2.5	Rapid Innovation	7
2.6	End-User Adoption	7
2.7	Framework as a Service	8
3	Microsoft Dynamics CRM Architecture Overview	9
3.1	Multi-tenancy	9
3.2	Metadata	9
3.3	Web Services	10
3.4	Data Model	11
3.5	User Interface	13
3.6	Business Logic	15
3.7	Workflow	16
3.8	Performance & Scalability	17
3.9	Security	18
3.10	Reporting	19
3.11	Mobile CRM	20

1 Executive Summary

As organisations strive to do more with less, many are exploring techniques to maximise productivity while reducing costs. While leadership mandates focus on enabling and empowering staff to deliver results, the challenge of driving organisational change and ensuring efficient operations remains of paramount importance. CIOs must provide support for a rapidly increasing number of systems, with increasing responsibility for the success of the organisation. All must be done in a unique environment with constantly changing requirements.

By taking advantage of Microsoft Dynamics CRM as a solution development framework, organisations can reduce project timelines without having to adapt the business to match the software model. Organisations can challenge the notion of business process change, while taking advantage of their current investment in Microsoft technology.

By using Microsoft Dynamics CRM as a solution development framework, organisations can invest once, yet develop multiple applications. Because you have a common layer for all of your Line of Business applications, you never have to repeat the process of building a framework. Thus, the cost of your future applications is drastically reduced, paying back the cost of your initial investment. Every time you add an application on top of Microsoft Dynamics CRM, you've reduced your base cost via the framework savings on the new application.

Cost savings also come in terms of time and quality. Custom business applications are often not extensible or scalable over the long term. They are developed to address a specific need but, due to resource constraints, are typically built in such a way that they cannot be quickly or easily repurposed or extended when new needs arise. With a solid, reliable framework, you have a predictable method for extending and scaling applications.

Lastly, you save money because Microsoft takes on much of the work for you. Instead of having to invest in updating and modifying your framework as new needs arise or technology changes, Microsoft takes on that cost of a full-time development and test team dedicated to updating the framework and making sure that it works with other products, such as Microsoft SQL Server, Microsoft Exchange Server, Microsoft Office SharePoint Server, and Microsoft Office. To take advantage of the latest technologies, all you need to do is upgrade Microsoft Dynamics CRM.

2 Microsoft Dynamics CRM as a Solution Development Framework

2.1 Introduction

Choosing a framework isn't just about solving today's problems. It is about building for the future. The framework you choose will be with you for some time to come, so you need to make sure you have a solid foundation on which to build your business, one that is able to respond as the business changes.

It is critical these days that businesses respond quickly to change. Therefore, organisations prefer not to spend years developing a line of business application. Solutions need to be implemented in months, not years. In today's economic climate, organisations can no longer afford to spend a year rolling out a CRM system, or any other kind of business application. If you spend six months trying to figure out what to build and another six months building it, the business you deployed it for will have already changed. You need to deploy quickly, but still need to develop a solid, usable, and fully-featured system.

It is also important to get the application out there quickly so you can get people using it and get their feedback. Then you can change, improve, and release again. That kind of development allows your organisation and your users to be productive more quickly, to develop more usable systems, and gets your users invested in the tools and their success. Selecting a framework that allows for rapid development and an iterative process is one of the keys to building a successful long-term strategy. The right framework helps ensure business agility.

Microsoft Dynamics CRM has grown in popularity with all types of businesses and organisations because of its ease of use and effectiveness as a traditional customer relationship management application. However, Microsoft Dynamics CRM encompasses much more, and is also a framework for easy-to-use, flexible, innovative, and collaborative business solutions that can be created more quickly and cost-effectively than if they were built using traditional development approaches.

Whereas organisations previously had to compromise application functionality or defer specialised organisation-specific functionality to a later funding cycle, with Microsoft Dynamics CRM they now can consider building new solutions that map far more closely to business needs from the initial stages of deployment.

This is significant because such solutions often deliver the greatest business value according to the organisation's specific, unique, and innovative processes, and enabling them sooner via the solution drives a more rapid rate of return on investment. The potential benefit of creating a cost-effective business solution based on Microsoft Dynamics CRM as a development framework is significant, but an in-depth understanding of the framework is required for organisations to gain the greatest advantage from it.

2.2 Lower Cost

Many organisations have moved to adopt technology based on the Microsoft .NET Framework because its reusable components and automated tools cut development time and cost. But while the .NET environment facilitates custom development, it doesn't eliminate the need for it. Simply stated,

the generic .NET environment does not have the rich business framework and architecture that accelerates industry-specific or line-of-business solution development.

Building completely from scratch is a costly prospect. You have to design everything from the ground up and build it out. Building the security system incorrectly means opening your company to attacks. Building a user interface framework that accommodates all the future possibilities is another challenge. And once it's built, you have to maintain it, update it, and of course, extend it. Many organisations endeavour to build a framework that can be reused for future applications. This is good in theory, but rarely occurs in practice. It is difficult and expensive to build a general-purpose framework. Furthermore, developers leave and the architects of the original framework may not be around when you go to build your second or third application. When that corporate knowledge departs, it can be time-consuming and costly to replace. Selecting an external framework settles the debate. You've got a solid, all-purpose framework that will be maintained by a dedicated group of external developers.

Microsoft Dynamics CRM is constructed with the building blocks necessary to jumpstart .NET-based business solutions development. Microsoft Dynamics CRM provides the common application components such as security, data access, workflow and presentation logic in a familiar development environment; Microsoft Visual Studio and languages such as C# or Visual Basic. Developers can apply the tools they already know to extend the application components in order to incorporate an organisation's business innovation.

Building less of the required functionality from scratch saves time and money, and, in our experience, an organisation using Microsoft Dynamics CRM as a framework on a large scale can realise savings that range from 50 to 70 percent of a traditional custom development project budget.

While the Microsoft Dynamics CRM framework provides building blocks that jumpstart development, enterprise organisations that choose this approach are not limited to using those building blocks alone. The ability to create tailored solutions that reflect specific business processes remains, because the Microsoft Dynamics CRM framework is designed to be extended with organisation-specific business logic, such as validation rules or unique workflows that an organisation can use to manage sales and marketing and customer service processes.

2.3 Faster Development

By taking advantage of Microsoft Dynamics CRM framework capabilities, projects can easily be approached in an iterative development cycle where core functionality is first delivered in a production-ready pilot phase, and additional functionality is quickly and easily delivered in subsequent phases of the project.

Using this approach, a typical pilot phase for a solution built on Microsoft Dynamics CRM can be delivered in one to two months, compared to a minimum of three or four months for a comparable pilot developed in .NET with purely custom components. The time savings for developing a full-scale application, which might take a year or more in custom development, are similar. The business benefit of this faster development goes beyond the associated budget savings. For example, it enables more frequent development updates, which offers a flexibility that enhances the effectiveness of solution deployments. This is especially valuable in environments where business requirements can change frequently with little warning.

The cost of ignoring such changes may be that the new capability is already outdated upon implementation. Alternatively, making changes during typical development cycles may lead to project delays. The faster time to benefit in using applications based on Microsoft Dynamics CRM as a solution framework means that mid-term course corrections can be more easily accomplished, which can support the actual speed of business change in dynamic and competitive environments. Thus, when the solution is deployed, it can more fully meet the organisation's needs and wants, regardless of whether those have changed throughout the development term.

2.4 Leveraged Upgrades

One of the biggest hurdles facing business applications is keeping up with advances in technology and new approaches to user interaction. Developing on the Microsoft Dynamics CRM framework allows the application to leverage these features from the framework itself as they become available. The metadata architecture of the framework makes implementing the upgrades practically seamless, since most of the core application is configured on the framework versus developed in framework - specific code. As a result, as the framework evolves and provides advanced capabilities, applications built upon the framework will inherit those capabilities.

For example, with the latest release of Microsoft Dynamics CRM various improvements were introduced, including multi-tenancy, a multi-lingual user interface and a re-architected workflow engine based on the Windows Workflow Foundation. Applications developed on the previous release of Microsoft Dynamics CRM were able to acquire these capabilities simply by upgrading and without the need for any additional development.

2.5 Rapid Innovation

Looking beyond the date when an organisation first deploys the application to the end-user community, faster development is also cost-effective because it facilitates iterative innovation and continuous improvement. Interface changes or business logic updates that might take days or weeks to implement with custom development can be made in a few hours.

As a result, updates can be made more frequently, so the application will enable the organisation's approach to ongoing innovation and adoption of best practices. That extends the application's useful life and increases the organisation's return on its investment compared to the outlay required for custom development. Subsequently, organisations gain from faster and easier maintenance and enhancement of their solution well into the future.

2.6 End-User Adoption

Of course, no solution really solves a business problem if people fail to adopt it. The user-adoption rate for solutions built on Microsoft Dynamics CRM is high, in part because of the framework's integration with the Microsoft Office software that millions of people already use every day as they do business. For example, Microsoft Dynamics CRM applications can use Microsoft Outlook as their user interface, so people can access them intuitively and confidently.

Additionally, out-of-the-box integration with Microsoft Word provides an easy mail-merge capability, and Microsoft Excel offers a quick data-analysis tool. With the multiple user interfaces and devices

supported, having access to customer information through the Web or through mobile devices further expands the usability of Microsoft Dynamics CRM-based solutions.

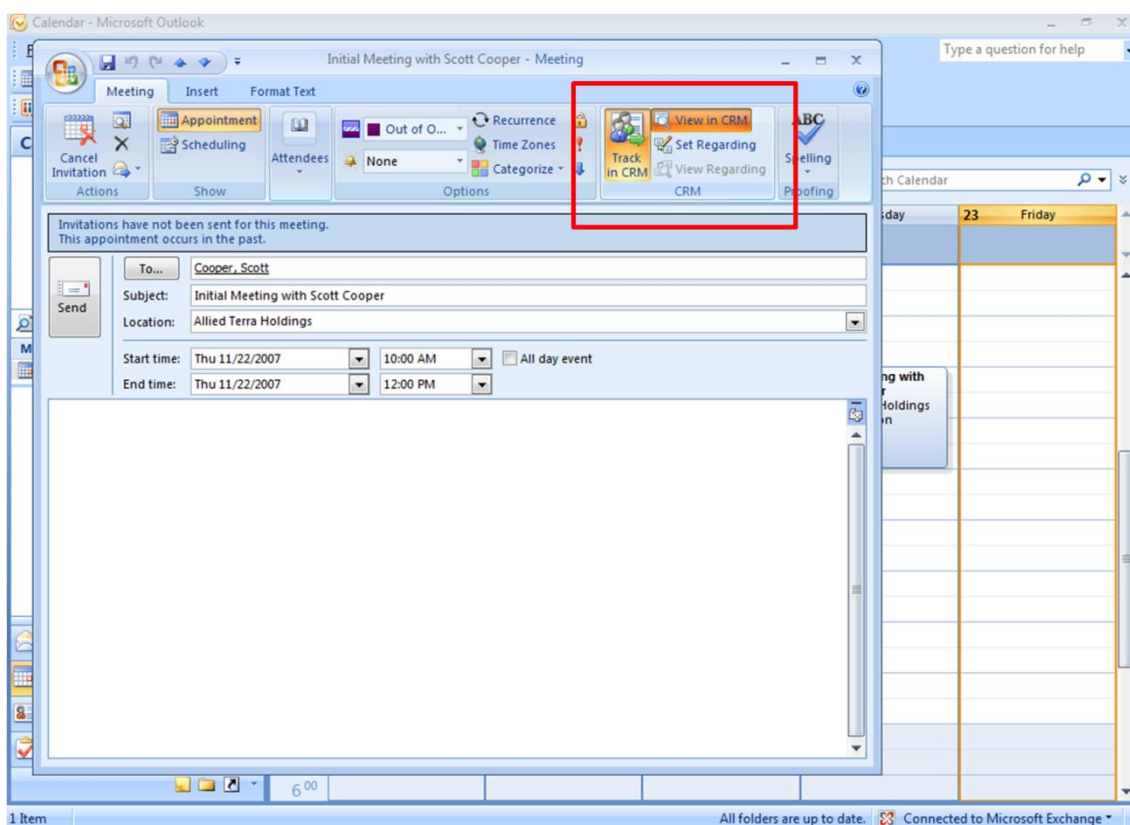


Figure 1 - Full CRM integration with Microsoft Outlook

That makes these applications as easy for end users to access and use as they are for developers to design, create, and configure, given the time and cost savings that come from deploying these solutions in service oriented architectures.

2.7 Framework as a Service

Microsoft Dynamics CRM is the framework that the Microsoft development team used to build the CRM application that you see. Microsoft's CRM development team uses the same framework and the same Software Development Kit (SDK) as our customers. This is not the case with most other CRM vendors. A more typical approach for vendors that build CRM products is a closed or black box approach in which the product is self-contained. When customers inevitably come back wanting to extend and hook into the system, the CRM vendor creates a public SDK that provides access to a limited subset of the functionality of the core product. In many cases the CRM vendor's developers don't even use the public SDK.

In the case of Microsoft Dynamics CRM, the SDK you use is the exact same web services SDK that is used internally by Microsoft's developers. And, of course, the framework is the same. If the product out-of-the-box is not what you want it to be, it is infinitely modifiable and extensible at virtually any layer. You are not constrained to Microsoft's vision of CRM.

3 Microsoft Dynamics CRM Architecture Overview

3.1 Multi-tenancy

To decrease the cost of delivering the same application to many different sets of users, an increasing number of applications are multi-tenant rather than single-tenant. Whereas a traditional single-tenant application requires a dedicated set of resources to fulfil the needs of just one tenant, a multi-tenant application can satisfy the needs of multiple tenants (divisions or departments within an organisation, etc.) using the hardware resources and staff needed to manage just a single software instance.

Tenants using a multi-tenant service operate in isolation from one another. Tenants can use and customise an application as though they each have a separate instance, yet their data, customisations and business logic remain secure and insulated from the activity of all other tenants. The single application instance effectively morphs at runtime for any particular tenant at any given time.

Multi-tenancy is an architectural approach that pays dividends to both organisations and end-users. Operating just one application for multiple divisions or departments yields tremendous economy of scale for an organisation. Only one set of resources is necessary to meet the needs of all users, a relatively small, experienced administrative staff can efficiently manage only one stack of software and hardware, and developers can build and support a single code base on just one framework (operating system, web server, database, etc.) rather than many. The economics afforded by multi-tenancy allow an organisation to offer the service at a lower cost to multiple divisions or departments.

Each Microsoft Dynamics CRM tenant deployed as part of a multi-tenant service is associated with a unique Microsoft SQL Server database, containing both data and metadata. So a database server hosting five tenants would have five separate databases, and each database would be configured according to the needs of the associated tenant. This allows for true separation of data among the tenants while providing high availability and fault tolerance at the database level using standard Microsoft SQL Server capabilities such as geo-clustering and database mirroring.

3.2 Metadata

Multi-tenancy is practical only when it can support applications that are reliable, customisable, upgradeable, secure, and fast. However, multi-tenancy also introduces its own challenges:

- How can a multi-tenant application allow each tenant to create custom extensions to standard data objects and entirely new custom data objects?
- How will tenant-specific data be kept secure in the database so one tenant can't see another tenant's data?
- How can one tenant customise the application's interface and business logic in real time without affecting the functionality or availability of the application for all other tenants?
- How can the application's code base be patched or upgraded without breaking tenant-specific customisations?
- How will the application's response time scale as multiple tenants subscribe to the service?

It is difficult to create an application that can meet these and other unique challenges of multi-tenancy. Inherently, a multi-tenant application must be dynamic in nature, or polymorphic, to fulfil the individual expectations of various tenants and their users.

For these reasons, Microsoft Dynamics CRM was designed to use a runtime engine that generates application components from metadata—data about the application itself. In Microsoft Dynamics CRM there is a clear separation of the compiled runtime engine, application data, the metadata that describes the base functionality of an application, and the metadata that corresponds to each tenant's data and customisations. Everything exposed to developers and application users is internally represented as metadata. Forms, views reports, workflows, user access privileges, web-services and tenant-specific customisations are all abstract constructs that exist merely as metadata. These distinct boundaries make it possible to independently update the framework, modify the core application, or customise tenant-specific components, with virtually no risk of one affecting the others.

Attempting to weave multi-tenancy throughout the fabric of an application's core logic and its underlying infrastructure is a complex undertaking. Building metadata-driven, multi-tenant applications from scratch without any prior experience is destined to be a time consuming and error-prone effort. In the end, many organisations struggle to succeed in building multi-tenant applications and end up wasting valuable time that could have been spent focused on the innovation of core business functionality and features. One problem is that traditional application development frameworks and platforms are not equipped to handle the special needs of modern applications. As a result, new types of frameworks are emerging to help simplify the development and deployment of multi-tenant applications and Microsoft Dynamics CRM is one of the first and most mature general purpose, multi-tenant, application development frameworks available today.

3.3 Web Services

Microsoft Dynamics CRM provides a dynamic web service interface for applications to access and manipulate CRM data and metadata and to interact with CRM business logic. The services exposed by Microsoft Dynamics CRM are WS-I Basic Profile Version 1.1 compliant (<http://www.ws-i.org/profiles/basicprofile-1.1.html>). This compliance support makes these services interoperable with both Microsoft and non-Microsoft development environments such as Microsoft Visual Studio, Eclipse or Oracle JDeveloper.

The entire framework is exposed through web services, which is the exact same collection of web services that Microsoft used in building Microsoft Dynamics CRM. This allows interaction with entities that ship in the box as well as with any custom entities that are created during configuration of the system. In addition, the Microsoft Dynamics CRM Client for Outlook has a local copy of the web services and programmability against these services is the same whether the user is connecting to the actual server or to the local data store on an offline client.

Web services in Microsoft Dynamics CRM are multi-tenant aware and completely metadata-driven, and data model modifications are automatically represented in the dynamically-generated web services. As you customise the data model of a particular tenant by adding custom entities, custom attributes or entity relationships, the web services for that tenant are dynamically adjusted. This provides a more consistent interface that is based on the framework metadata. In addition, developers have one technique to learn in order to manipulate any of the framework entities both built-in and custom.

Applications can leverage these web services to integrate with other environments. For example, applications can leverage APIs to access data in other systems, build "mashups" that combine

information originating from multiple data sources, include external systems as part of an application process, or build fat clients to interact with the framework data.

In addition to a web service supporting the traditional create, retrieve, update and delete (CRUD) operations for system and custom entities, Microsoft Dynamics CRM includes a web service that allows developers to interact with the underlying metadata. This enables organisations to integrate metadata with enterprise application lifecycle management (ALM) tools such as Microsoft Visual Studio Team System. Organisations can build reusable metadata templates that can be adapted and extended to create many unique applications within a multi-tenant environment. For example, an organisation may wish to deploy an organisation-wide template as a base configuration for all applications, including standard schemas, business logic, security roles, integrations and workflows. This base-template could be merged with a division-specific template, which in turn could be merged with a department-specific template and so on. This approach can help to accelerate significantly the deployment of new applications.

3.4 Data Model

Microsoft Dynamics CRM uses the concept of an entity to describe its data storage container. If you were to think about it in common database terms, an entity would be equivalent to a table in a database. Each entity will have several attributes that will be used to describe and store individual fields of data. Continuing the database analogy, these would be the same as columns on tables.

Business data is the cornerstone of any good line of business application. A number of standard data entities that are commonly known to exist in business applications built on the framework are pre-defined out of the box. Common entities like the Contact or Account are the type of expected data entities that you would find. These are referred to as system entities and they exist so that common scenario implementations don't require building of these often used entities. By allowing the framework to manage the data model, Microsoft Dynamics CRM is able to provide a rich set of features like de-duplication, offline synchronisation and a full web service interface even to custom entities. It would take extensive development work to build each of these features from scratch.

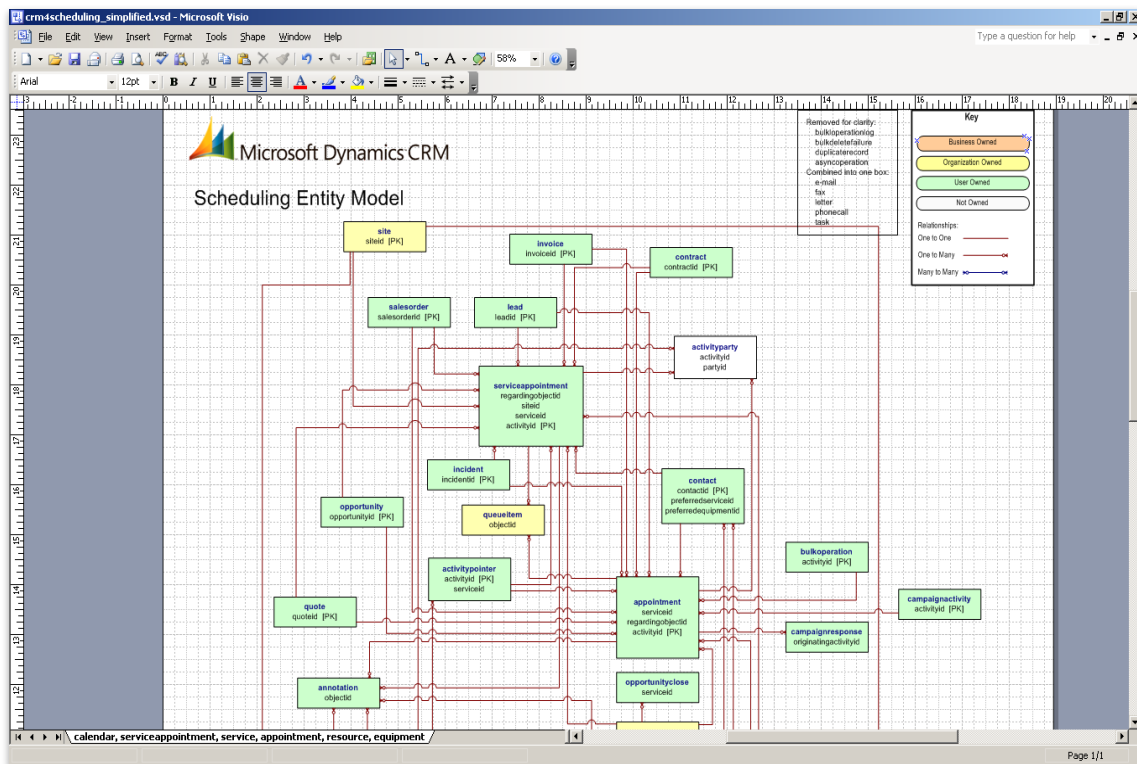


Figure 2 – Out of the box CRM data model for the most common requirements

Many of these system entities are customisable, meaning the implementer of the CRM solution can rename an entity's display labels, rename attribute display labels, and even add additional attributes appropriate for specific solutions. An example of this is renaming the system entity Account. Depending on the solution this could be Client, IFA, or Underwriter for example. By allowing the developer to change the display name, including all associated user messages so that the user never sees the default name is incredibly powerful. It helps ensure that the users feel that the solution is customised for them, and not just one-size fits all.

To provide flexibility, in addition to the customisation of the system entities, the framework also allows you to define completely custom entities. This allows creation of entities to support the needs of any solution either vertical or horizontal in nature. These custom entities are first class entities available for use with the entire extensibility capabilities of the framework.

In order to allow modelling of real world applications, Microsoft Dynamics CRM uses the concept of relationships. Relationships define how one entity relates to another as well as to specifying the characteristics and behaviours of those relationships. Continuing the database analogy further, these would be the same as primary and foreign key relationships between database tables.

The framework uses relationship types to further define how relationships work in the system. For example relationships can be system, parental or referential. The relationship type establishes how the framework should handle operations that could affect the entities that make up the relationship. For example if you delete a contact you wouldn't want to delete the associated company. That would be an example of a referential relationship. Alternatively, all the notes associated with that contact would no longer be relevant after the contact was deleted so a parental relationship is appropriate.

Changes to existing system entities and the creation or modification of custom entities can be accomplished using either the user interface or the metadata web service. A developer is able to define and modify entities and does not need to be a database professional with extensive database skills. The framework then takes care of the data and metadata management behind the scenes and provides access to the data using its rich web service SOA interface or via the user interface for interactive users.

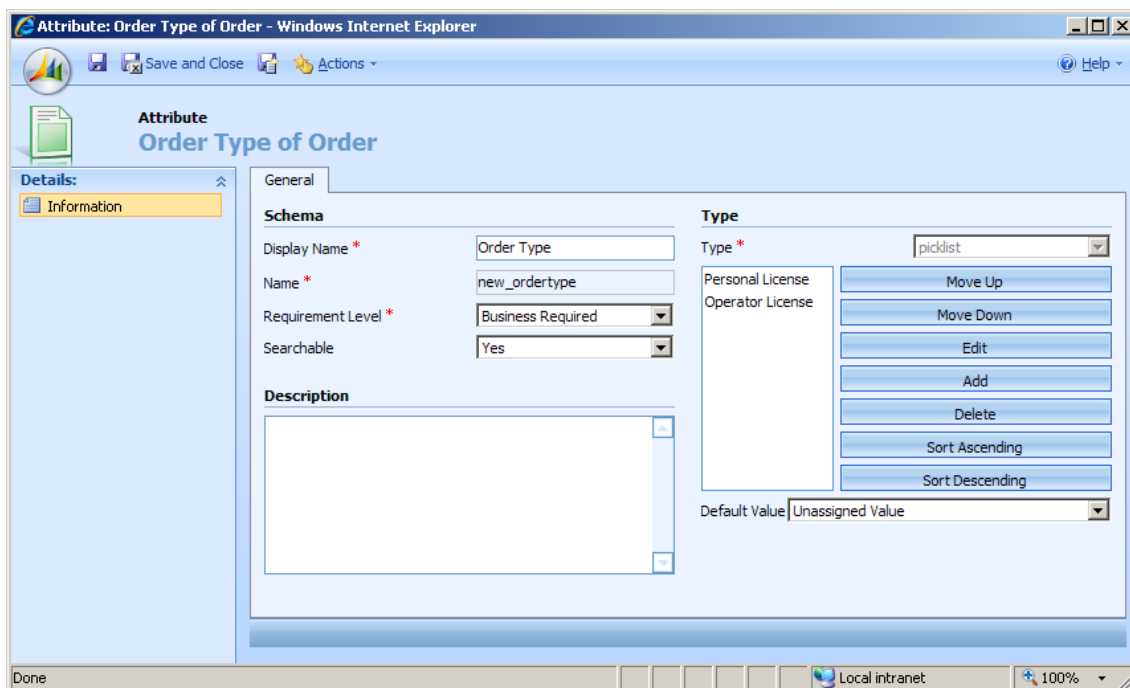


Figure 3 - Defining a custom attribute using the CRM user interface

One of the core capabilities of Microsoft Dynamics CRM is the ability for data to be synchronised with a client machine in order for that data to be available even when the client machine is offline. System entities, custom entities and their associated data can be synchronised by a user using the Microsoft Dynamics CRM Client for Outlook. Microsoft Dynamics CRM manages the synchronisation of both data and metadata; including any changes the user makes while offline. Using either the user interface or the metadata web service, you can identify which entities are available offline and how much of a specific entity users take offline.

3.5 User Interface

Once the data model has been established the next logical step is to allow users to input, modify and review data associated with the entities defined. Microsoft Dynamics CRM provides the user experience using HTML-based pages to browse and manage data. You have the ability to customise many aspects of the user experience including form content and site navigation. Interaction with the user can also be customised by events exposed by the user interface and custom scripting. External applications can also be integrated as part of the navigation as well as embedded inside CRM forms.

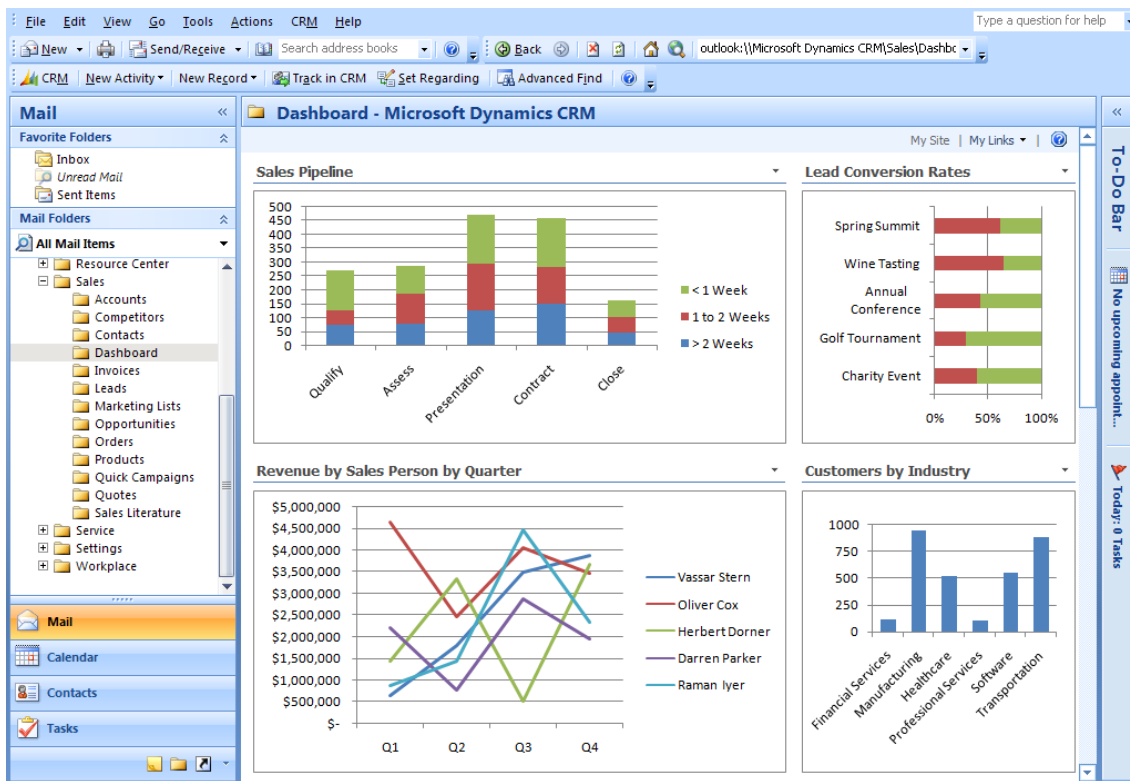


Figure 4 – An external SharePoint dashboard integrated into the CRM user interface

Customising forms is quite simple as the framework has provided a number of useful and flexible objects. The entity forms contain a set of tabs and you can also add additional tabs. Each of the tabs contains one or more sections. You may control and configure the layout of the sections within the tabs. Each section can contain one or more of the entity attributes or an HTML IFrame for external content integration. The order and layout of the attributes inside the section can be customised using the form editor within the user interface.

As part of the entity form the framework also provides the ability to run custom JavaScript when events occur both at the entity form level and field level inside the form. The custom scripts that are triggered by the events can perform tasks like setting default values, performing client side validation or calling out to 3rd party services.

It is unlikely that any two tenants built on the Microsoft Dynamics CRM framework will have the same demands for site navigation. The navigation model supports this need by providing the ability to integrate other web applications and to modify the default navigation. This includes the ability to place custom entities into the navigation hierarchy in specific locations.

Another popular approach is for organisations to use the web service interface to build completely custom applications that interact with the framework. For example, an organisation could build a rich client application leveraging Windows Presentation Foundation (WPF) that provides a different look and feel for the user experience but under the covers uses the framework web services.



Figure 5 - A custom WPF application built on top of the CRM framework

3.6 Business Logic

One of the challenges of building line of business applications on top of an off the shelf framework is the difficulty of integrating custom business processes into the heart of existing framework processes. Microsoft Dynamics CRM addresses this problem with support for Plug-ins and a robust Workflow capability. These two capabilities are implemented using a unified event framework.

Plug-ins provides the opportunity for organisations to insert custom code, written in a .NET language such as C# or Visual Basic, into framework services and takes control before or after a framework operation is performed. Take the example of a user adding an account record. When that user presses the save button, the user interface will submit a "Create" request to the framework. Using Plug-ins it is possible to register logic to be invoked before the account is created in the database. This allows the plug-in to modify any data, perform more complex validation, reject the operation or any other value added processing before the "Create" request was processed. You can also register to be notified after an operation is completed, which might be appropriate for auditing or integration with external applications, for instance.

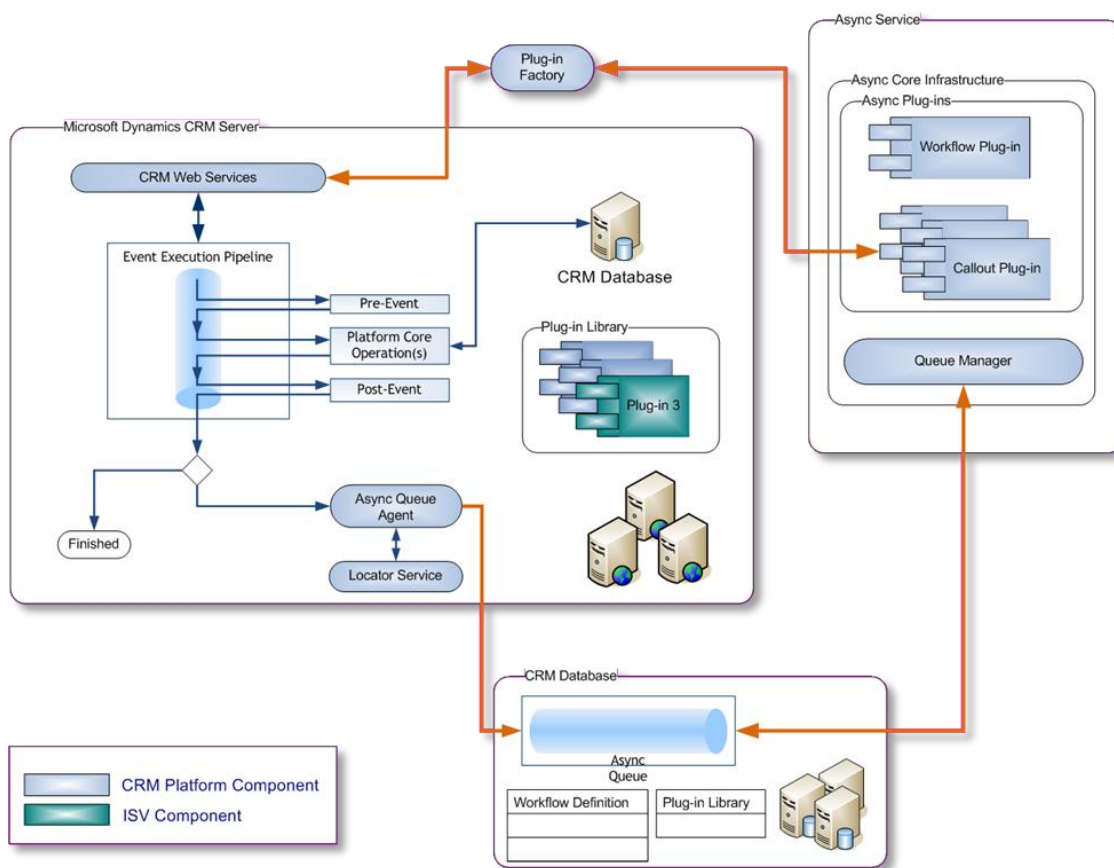


Figure 6 – Unified event framework for both synchronous plug-ins and asynchronous workflows

A plug-in has the ability to execute both on the server when requests are made or they can be provisioned to run offline. Offline support is used to facilitate having custom logic run when a user is using the Microsoft Dynamics CRM Client for Outlook and is offline. Imagine building an offline application and you need to implement custom validation that can't be performed using the form customisations. When the user is back online and able to synchronise, would it be a good experience for that user to receive error messages for transactions entered hours or days earlier? By marking the validation plug-in as available offline, the user would get that error immediately even when they are offline because that plug-in would fire in the offline client.

3.7 Workflow

Windows Workflow Foundation is integrated with the Microsoft Dynamics CRM framework to provide a robust engine to implement business process automation. You can implement workflows that are simple (a single step) or complex (a series of steps, checks, waits and rules that control the automation).

Workflows can be executed manually, when records get created or deleted, or even when a particular data value in a record changes. You can use workflows to automatically create activities, update records, send email on a user's behalf, and extend workflows with your own custom workflow actions.

Workflows are different from plug-ins because they are more visible to the user. A Plug-in executes behind the scenes and unless they throw an error the user never knows they exist. Workflows on the other hand are very visible to the user through the user interface. The user can see when they have

run, if errors occur and can even manage the workflows. The user may configure workflow options, including when they should run and provide parameters that control how they work. Through the user interface, users are able to view a list of workflows that they have access to as well as define new workflows if they have been granted the appropriate security.

Workflows can be defined at the organisational level to apply to everyone, or by individuals as personal workflows. For example, when new leads come in, one user can set a workflow to automatically send an email using a template that user created, while another user can have a workflow automatically schedule a phone call activity for the user to call the new lead.

In addition to the list of steps available out of the box, developers can create custom workflow activities which can be added to the lists of available steps. Custom activities are written in a .NET language such as C# or Visual Basic.

The screenshot displays the CRM workflow configuration interface. At the top, the 'Workflow Properties' section includes fields for 'Workflow Name' (Close Task - New Business or New Card H...), 'Entity' (Task), and 'Publish As' (Workflow). Below this, the 'Available to Run' section has checkboxes for 'On demand' and 'As a child workflow'. To the right, the 'Options for Automatic Workflows' section includes a 'Scope' dropdown (Organization) and a 'Start when' section with several checkboxes: 'Record is created' (checked), 'Record status changes', 'Record is assigned', 'Record attributes change', and 'Record is deleted'. A 'Select' button is next to the 'Record attributes change' checkbox. The main area is a step editor with a tree view showing conditional logic: 'If Task:Subject equals [Check BDI for prospect], then: ... Otherwise, if Task:Subject equals [Check new Card Holder], then: ... Otherwise: ...'.

Figure 7 - Defining workflow through the CRM user interface

3.8 Performance & Scalability

Microsoft Dynamics CRM optimised, metadata-driven architecture delivers extraordinary performance, and scalability for on-demand, multi-tenant, line of business applications.

Microsoft Dynamics CRM takes unique advantage of Microsoft Windows and Microsoft SQL Server to provide enterprise levels of performance and scalability. The application is message-based and stateless, allowing for easy horizontal scaling through standard network load balancing methods. The application can be tuned for performance and scalability using common database tools and skill sets, and is designed to maintain the benefits of tuning as the application changes.

Because metadata is a key ingredient of Microsoft Dynamics CRM, the framework's runtime engine must optimise access to metadata; otherwise, frequent metadata access would prevent the framework from scaling. With this potential bottleneck in mind, Microsoft Dynamics CRM uses metadata caches to maintain the most recently used metadata in memory, avoid disk I/O and improve application response times.

In benchmark testing, Microsoft Dynamics CRM demonstrated the ability to scale to meet the needs of an enterprise-level, mission-critical workload of 50,000 concurrent users while maintaining performance at sub-second response times. Tests also demonstrated that Microsoft Dynamics CRM is capable of achieving sub-second response times running user transactions against a database of over 1 billion records, containing more than 1 Terabytes of data.

3.9 Security

Security plays an important part in any framework that you might consider using as a foundation for building applications upon. At the most basic level security is needed to maintain access control over data and services, granting access only when appropriate. Denying access is pretty straight forward, but building a security model that is flexible enough to grant access to users and to span departments, divisions and different users' roles is a lot more complex. Microsoft Dynamics CRM provides a rich security model that applies to the built-in application and framework components but also to the customisations you build as part of a solution

For single sign-on, authentication and authorisation across the multiple tenants, Microsoft Dynamics CRM leverages Microsoft Active Directory. This enables a user to enter and exit multiple tenants that they have been granted access to without having to log in multiple times. While the user is within a tenant they will not be able to view the information in another tenant even if they have been given access to that tenant. They will need to exit and re-enter the other tenant to have access to the data. In this fashion the framework is able to guarantee that the data is not combined in a fashion that is not desired.

Once user access has been granted, the per-tenant security model provides a highly efficient means for establishing and changing security privileges to ensure that users always have the appropriate access. Role-based security defines privileges based on business roles within each tenant rather than on individual users. Users can have one or many roles associated with them with additive privileges across roles, so users can perform many roles within a tenant and the system will automatically recognise the correct security privileges.

To provide additional flexibility the concept of business units is used to group users and grant authorisation based on the roles. Each tenant has the concept of an organisation; the business units exist within the organisation similar to how you might envision divisions within a company. In fact business unit design can mirror or help facilitate company structure within the security model. Each business unit defines roles that are either inherited from the parent business unit / organisation

hierarchy or ones that are specific to their needs. Each user is then allocated to a single business unit, and assigned one or more of the security roles available to users in that business unit.

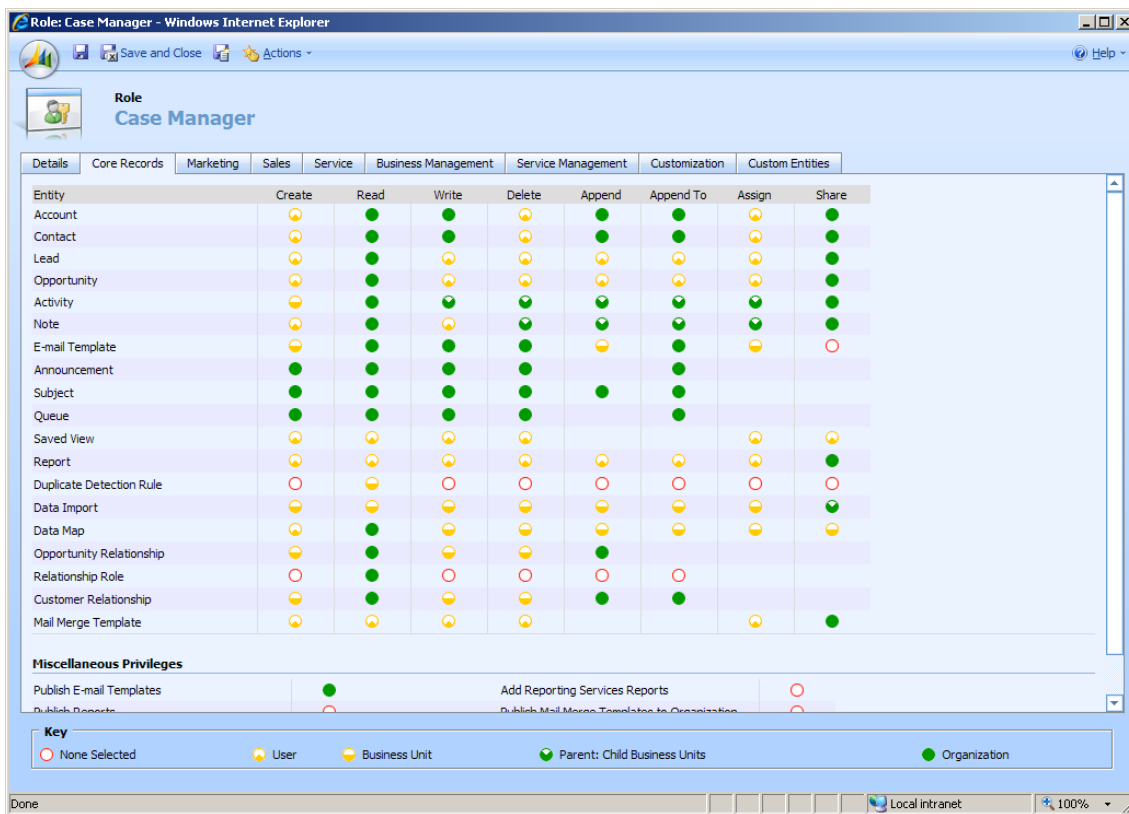


Figure 8 - Defining role-based security through the CRM user interface

3.10 Reporting

So far we have discussed how to solve business problems by customising, extending and automating business processes using the Microsoft Dynamics CRM framework. Through those capabilities organisations collect and manage data in a way that is natural for business users, and as a result users are more productive in their day to day tasks.

The next logical step is to figure out how to use this data to create reports and other analysis to facilitate faster more accurate business decisions. Microsoft Dynamics CRM leverages Microsoft SQL Server Reporting Services as the engine for producing and distributing reports. Not only can developers build custom reports using Microsoft Visual Studio, but end users can also use the report wizard to create custom reports as needed using the user interface. The framework then manages these report definitions and allows them to be scheduled to create snapshots or to be used when the user is offline.

Earlier we talked about the importance of application security and how the framework enforces it both through the user interface and the SOA Web Service interface. Security for reports is just as important but typically can be more challenging to implement as part of a solution. Microsoft Dynamics CRM uses a concept called filtered database views to provide that consistency in the security model. Filtered views are created and maintained by the framework metadata for each built-in and custom entity defined in the data model of the solution. They incorporate the security roles, business unit

design and record ownership to enforce what data users have the ability to run reports against. Additionally, since reports are represented in the data model by the report entity, reports themselves can be controlled using the framework security model to determine who can see, run and modify each report.

Microsoft Dynamics CRM provides a number of built-in reports that cover common scenarios that users will need reports on. These built-in reports are a great way for organisations to model their report writing as they can be used as starter reports. Often times the built-in reports are more complex than custom reports that are created so developers can use them to learn techniques and as a starting place for creating their own custom report. As organisations write and deploy custom reports as part of their solution they are shown in the report list alongside the built-in reports.

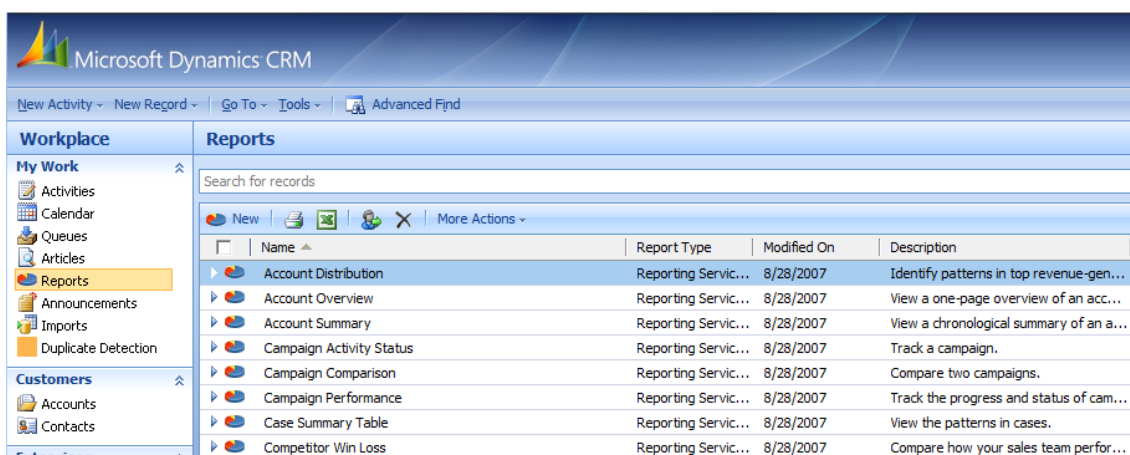


Figure 9 - Custom and built-in reports are shown in the CRM report list

3.11 Mobile CRM

Mobile Express for Microsoft Dynamics CRM delivers the power of the framework to your mobile workforce with ease. The Mobile Express client provides users the ability to view, create, and modify their CRM data on just about any internet capable mobile device. There is no installation required on the mobile devices to enable this functionality. Additionally, an administrative interface is available to enable organisations to control what and how users view their CRM data on the mobile device.

The client uses common metadata from Microsoft Dynamics CRM so users can access all built-in and custom entities within CRM and is optimised for limited bandwidth environments and will work on any HTML 4.0 Web browser and requires no client-side script or installation on the client device. Mobile Express for Microsoft Dynamics CRM also works with the multi-tenanted architecture allowing users to point at different tenants from the same device.



Figure 10 – TenDigits MobileAccess solution for Microsoft Dynamics CRM

Microsoft and Ten Digits Software Inc. currently have an alliance which delivers full-featured mobile access to both Microsoft Windows Mobile phones and RIM BlackBerry. MobileAccess gives organisations the choice and flexibility to meet the requirements of their users while giving them anytime, anywhere access to Microsoft Dynamics CRM. In addition MobileAccess is administered within Microsoft Dynamics CRM which helps minimise IT management overhead and setup.