

Platform

Microsoft Dynamics[®] CRM 2011

Building Business Applications with Microsoft Dynamics CRM 2011

A guide to Independent Software Vendors and Developers

November 2010

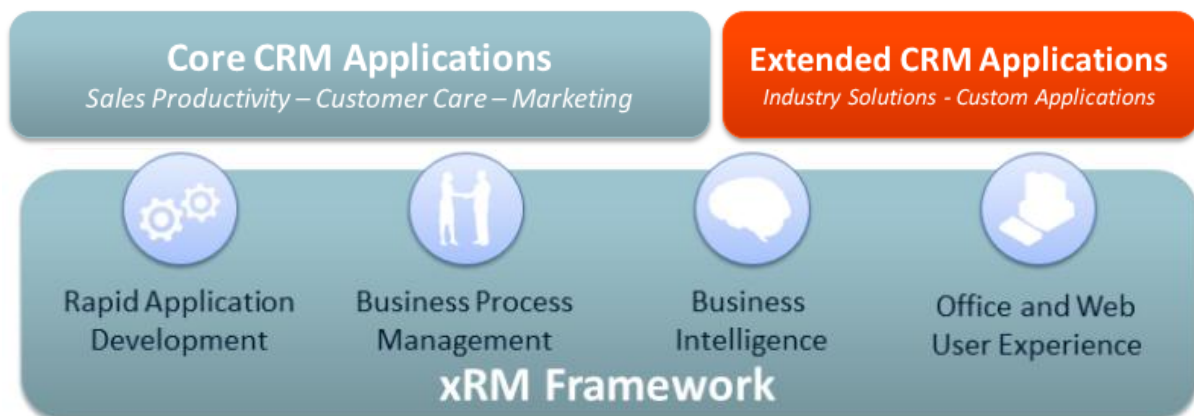


Contents

Introduction	3
Building Solutions	6
Modeling Business Data.....	10
Custom Activities	11
Connections	11
Field Level Security.....	12
Data Auditing	12
Building a Better User Experience	13
Role Based UI	16
Visualizations – Charts, Dashboards and Reports.....	17
Charts.....	17
Dashboards	18
Reporting	19
Integrating with SharePoint®	20
Dynamics CRM 2011 Programmability Overview	21
Business Processes in Dynamics CRM 2011	21
Workflows	22
Dialogs.....	23
Workflows and the Solution Framework	25
CRM Workflow Upgrade	25
Workflow Assembly Versioning.....	25
Working with Data	26
WCF Endpoint.....	26
Using the LINQ Provider	27
OData and the REST Endpoint.....	27
Plug Into the Sandbox	27
Event Pipeline and Windows Azure App Fabric Service Bus.....	29
Moving to Claims Based Security	30
Hosting Content in Windows Azure.....	30
Wrapping Up.....	31
Next Steps	31

Introduction

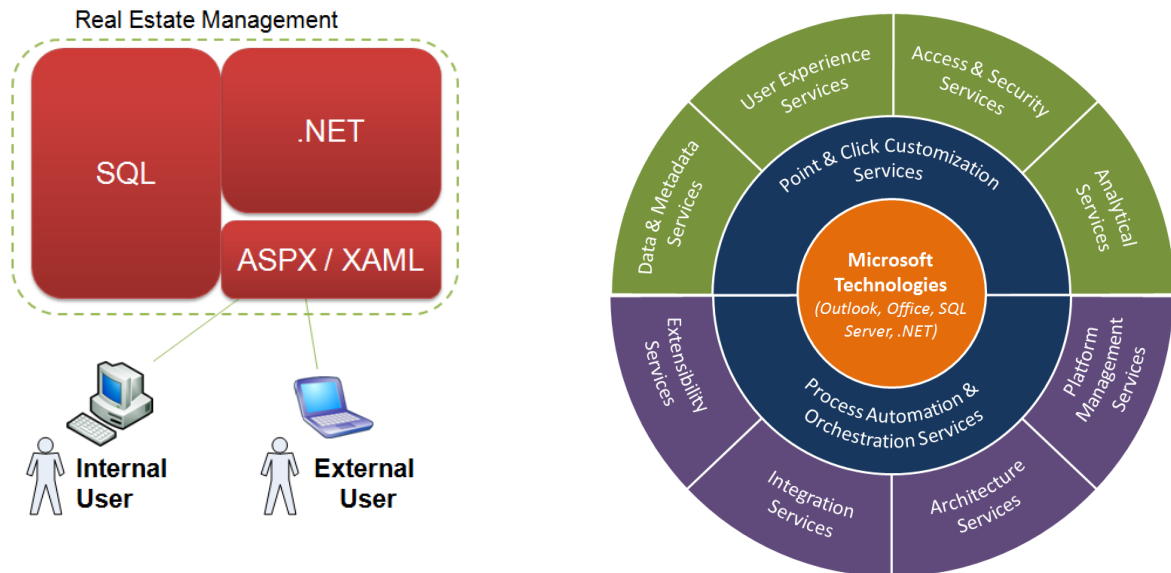
Microsoft Dynamics® CRM 2011 is a Customer Relationship Management system that is designed to empower business users in the areas of Sales, Service and Marketing. Microsoft® is leading with the Cloud and CRM is a key enterprise-ready piece of the Microsoft Cloud Services strategy. Microsoft Dynamics CRM 2011 builds on the foundation of prior versions of Dynamics CRM to offer independent software vendors (ISVs) a powerful platform for building line of business applications. These applications are often referred to as Extended CRM applications or xRM Framework applications because they leverage the relationship tracking capabilities beyond the typical customer relationship management (CRM) scenarios. If your business model is Real Estate Management, Fleet Management, Patient Management or Vacation Activities Planning to name a few, it can be implemented via an Extended CRM application in Dynamics CRM 2011. These days, line of business applications (LOB) model and track connections between various types of business data. Dynamics CRM 2011 provides for declarative development of relational business applications with flexible data models and dynamic services. ISVs building Extended CRM applications on Dynamics CRM 2011 use the .NET Framework™ and a variety of other common Microsoft platform technologies such as Windows Workflow Foundation (WF), Windows Communication Foundation (WCF) and SQL Server®. These technologies are assembled as part of the xRM Application Framework. The xRM Application Framework provides the following high level capabilities to simplify application development.



The xRM Application Framework is the common foundation used by both the Core CRM applications built by Microsoft and the Extended CRM applications built by ISVs. For example, an Insurance Agency or Finance firm could use Dynamics CRM 2011 in a traditional sense but could also benefit greatly with an Extended CRM application to manage policies, documents and interoperate with other industry standard applications. These applications all take advantage of the following high level features of the xRM Application Framework.

- Models encompassing multiple domains that automatically include data, presentation, workflow and security to name a few.
- Business application services that include extensible client experiences, multi tenancy, robust web services that all adapt to the current application's published model.
- Enterprise scalability and a platform that provides a proven commitment to backward compatibility and early adoption of key Microsoft technologies (for example .NET 4). The Cloud: We're All In – including Microsoft Dynamics CRM 2011 with choice of running in the cloud with CRM Online and interoperability with other Microsoft Cloud products like Windows Azure™.

To better understand the xRM Application Framework let's explore a Real Estate Management application and then think about the different components required to build a complete solution. The diagram below illustrates how the components of the xRM Application Framework can work together to meet these application requirements.



Extended CRM applications can be fast to build with point and click customizations and drag and drop UI designs. Applications can take shape in just a few hours or they can also be fully planned out over a longer period of time using more traditional project planning techniques. As needs reach beyond what is possible with point and click customizations the solution can be extended by developers with custom code as needed. Developers work with familiar tools like Microsoft Visual Studio 2010® to interact with the services and extend as needed. End users interact with the application using a familiar browser based interface or through the CRM client for Microsoft Outlook®. Generally, applications built using Dynamics CRM 2011 works the way users expect that they should. Using dynamic service capabilities, Extended CRM applications are able to adapt to the changing business needs. For example, when a new attribute is added to the model it is immediately available in the UI, from the Web Services and also available for reporting and workflows. In a traditional application this would have required quite some effort to accomplish. Customizations as well as full extended applications are packaged up as a "Solution", another new feature of Dynamics CRM 2011. This flexibility enables ISVs to build a common solution to a problem that can be further tailored and customized to fit the individual needs of their end customers in a more cost-effective way. ISVs can then publish their solution in the Dynamics Marketplace (<http://www.microsoft.com/dynamics/marketplace>) where customers can easily locate the solution.

The pre-assembly of the various popular Microsoft platform technologies that make up the xRM Application Framework allows ISVs to minimize the effort they expend to glue together all these components for use. For example, in a typical database application you would create a database table named *CommericalProperty* to represent a real estate property. A developer would then typically create a custom data access layer and a series of data input screens to support it. The xRM Application Framework provides out-of-the-box drag and drop user interface customization capabilities and a WCF web services endpoint for programmatically working with the data. If reporting or workflow was required to work on the data in the *CommericalProperty* table, a developer would traditionally write a bunch of reporting services or workflow specific code to build that into the

application and interface with the products. With xRM the reporting can be done by any user out of the box using the report wizard which sits on top of Microsoft SQL Reporting Services. The user gets the direct access and familiarity of the application and the power of SQL Reporting Services. The same is true if a workflow needs to be created as a workflow wizard is available directly with the xRM application. The built in form designer offers drag-and-drop design WYSIWYG (What You See is What You Get) customizations and modeling the attributes that make up the *CommercialProperty* table is similarly achieved through a simple, browser delivered, GUI interface. Outlook integration, customizable security roles and scalability are also all out-of-the-box items with Dynamics CRM. Dynamics CRM supports multiple currencies and languages in the same implementation as well as a model for multi-tenancy for ISVs to host multiple customers on a single set of hardware.

ISVs that have built on prior versions of Dynamics CRM benefit from a strong commitment to backwards compatibility across product versions. Prior versions as well as Dynamics CRM 2011 provide an upgrade path that allows ISVs to move to the latest version with minimal effort. For example, Dynamics CRM 2011 introduces a new WCF endpoint, but the prior Dynamics CRM 4.0 endpoint will still be fully supported. This allows applications written on the prior versions to move forward without extensive effort. Dynamics CRM 2011 is also moving to the Microsoft .NET Framework 4, which includes the latest workflow and communications APIs. Developers will have access to the latest programming language innovations and productivity as part of Microsoft Visual Studio 2010. Developers are shielded from much of the effort involved in upgrading these platform components to these latest versions.

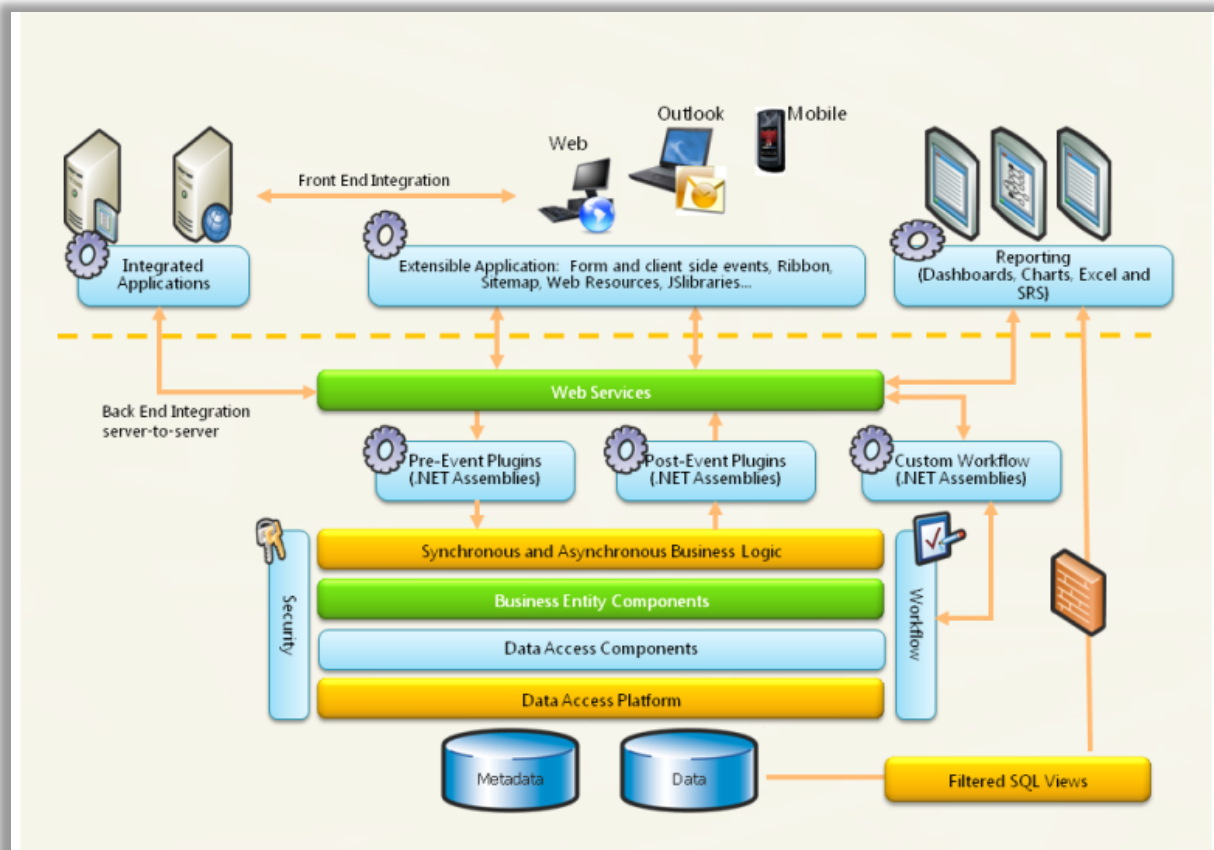
Today, ISVs are building solutions to target both horizontal and vertical solution opportunities. Extended CRM applications have been built for industry verticals like banking, insurance, public sector, professional services, health care and others. Often times the one platform can also fulfill the needs of multiple applications providing not only reduced cost but a more integrated solution. In addition many ISVs have built horizontal solutions such as enhanced marketing capabilities.

Deployment flexibility allows xRM solutions to be deployed on-premises, in partner hosted datacenters or in-the Cloud with Microsoft Dynamics CRM Online. The xRM Application Framework shields ISVs from most of the differences in these different deployments- an ISV can write their app once and allow a customer to deploy it into any of the supported environments. New sandboxing capabilities enable code and reports to now be deployed even in cloud deployments with a consistent level of isolation across tenants in their multi-tenant deployments.

Dynamics CRM 2011 includes a number of new capabilities that further enable rapid application development. These new capabilities in the xRM Application Framework are centered on developer productivity, power of choice and rich framework services. Included in these new features is native integration with the Windows Azure Platform. The Windows Azure Platform is an Internet-scale cloud computing and services platform hosted in Microsoft data centers. The Windows Azure Platform includes the foundation layer of Windows Azure as well as a set of developer services which can be used individually or together. Azure enables extending and connecting Microsoft Dynamics CRM 2011 applications with cloud based applications and services. This includes being able to publish events from Microsoft Dynamics CRM 2011 to the Windows Azure AppFabric Service Bus with a simple registration.

The rest of this paper will dive deeper into the new capabilities offered by the xRM Application Framework for building Extended CRM applications.

The following diagram illustrates some of the key components that will be discussed and illustrates at a conceptual level how they interact. These components rely on Windows Server® and Microsoft SQL Server for data storage.



Building Solutions

The ability to customize the xRM Application Framework enables ISVs to build a wide variety of Extended CRM applications. The customizations can be completed by a team consisting of both developers and non-developers with a common goal of building a complete application or product. Prior to Dynamics CRM 2011 these customizations could be moved from implementation to implementation but there really wasn't a container that packaged them together- each extension point of the framework required a different approach to redistribution. Additionally, versioning an application's customizations and interactions between different ISV customizations were difficult and situations arose where different vendor's products conflicted with each other by making changes to the same elements.

Microsoft Dynamics CRM 2011 introduces a new concept called a "Solution" to address these challenges. Solutions often represent some level of business functionality that when installed and used in a CRM organization will solve a specific business need. For example, a small solution might simply provide a way to verify postal addresses in a system. While larger solutions might contain an entire Real Estate Property Management application. Solutions can also be building blocks combined together to compose the complete business solution. In the current examples, the Property Management application could rely on the Address Verification solution to perform that specific service. Together though they would solve a larger business challenge of an organization. ISVs are

able to build and sell solutions that represent their products. These solutions can be published on the Microsoft Dynamics Marketplace where customers can easily locate the ISV solution. The marketplace is integrated with Microsoft Dynamics CRM 2011 allowing customers to navigate directly to the marketplace and checkout ISV solutions.

Solutions act as a container to hold the comprehensive set of components that make up an application's customizations. Solutions are authored, packaged and maintained as a single unit of software. Going forward each customization in the system will be done in the context of a Solution. Solutions can include a number of different types of components. The following are the supported solution components:

Schema	User Interface	Process/Code	Templates	Other
<ul style="list-style-type: none"> • Entities • Attributes • 1:N Relationships • N:N Relationships • Global Option Sets 	<ul style="list-style-type: none"> • Forms • SiteMap • Entity Ribbons • Application Ribbons • Views • Web Resources • Charts • Dialogs • Visualization Modules • Dashboards 	<ul style="list-style-type: none"> • Workflow Definitions • Plug-in Assemblies • SDK Message Processing Steps 	<ul style="list-style-type: none"> • KB Article • Mail-merge • E-mail • Contract 	<ul style="list-style-type: none"> • Reports • Connection Roles • Security Roles • System Settings

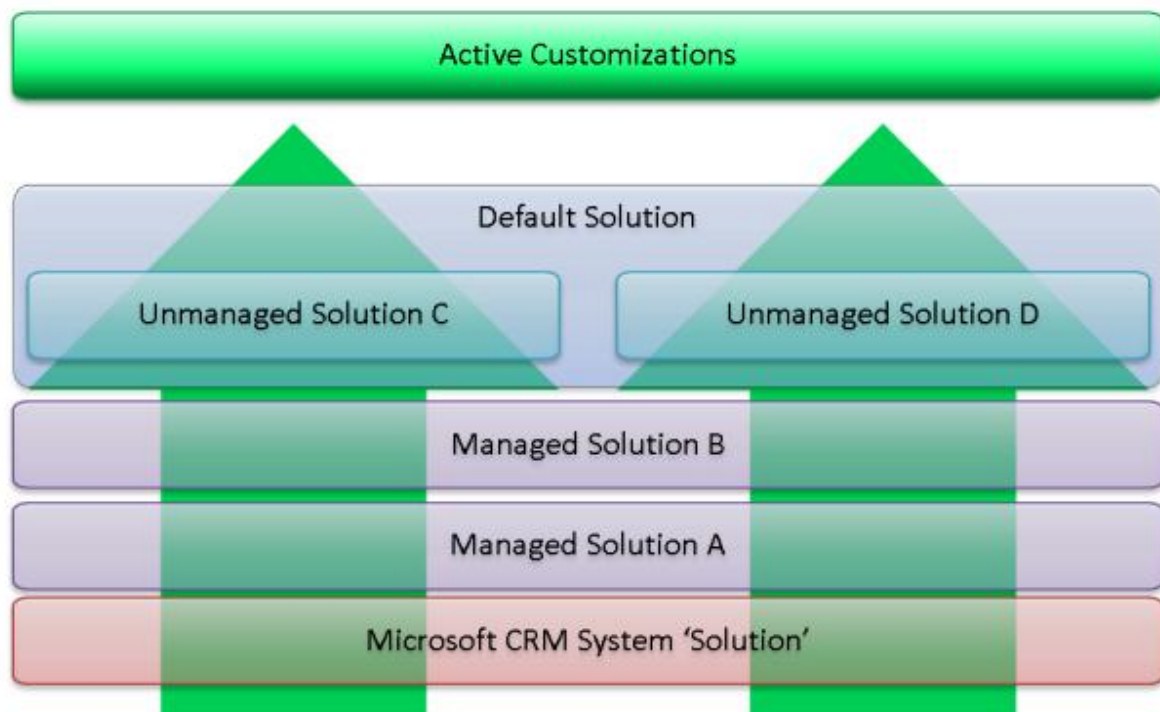
Each organization will have a solution named default created at the time of deployment. The default solution provides a reference to all of the components in the system. For basic deployments all customizations can be performed directly on the components in the default solution. This provides a very similar model to what existed in prior versions of CRM where there were no different levels of isolation among customizations.

The solution framework allows for solutions to exist in two states. A solution can be either unmanaged or managed. All unmanaged solutions can be modified with further customizations. You can create as many new solutions as you want. Each starts out as empty containers in an unmanaged state. Existing components from either the system or other managed solutions can be added as a component of another unmanaged solution creating a dependency. When an existing managed component is added it creates a reference to the component and not a copy. Multiple unmanaged solutions can contain a reference to and then customize the same components. For example, Solution A and Solution B could both add the existing entity Account to their solutions. Any change to a component in an unmanaged solution can be seen by all other unmanaged solutions referencing the same component. In this example, if a new attribute were added to Account in Solution A, that same new attribute would be seen on Account if you looked at Solution B. So basically, all unmanaged solutions co-exist in the same unmanaged layer of customizations.

To allow for isolation and for solutions being built by multiple ISVs to co-exist together and avoid conflict the solution framework also supports the concept of a solution being in a managed state. The managed state is a progression from the unmanaged state and is enabled by specification at the time

that the solution is exported from the source system containing the unmanaged version. In the managed state a solution's components can't be directly modified in the context of the managed solution. They can however, be added as references to a new unmanaged solution or as part of the default solution as we discussed previously. There are times where an ISV would like to prevent any further customizations of a solution's components when a customer uses it. Managed properties are another new feature of the solution framework allowing specification at the component level of what customizations can be performed on the managed component.

The following is a layer diagram representing how managed and unmanaged solutions might co-exist in a single deployed organization.



Notice in the diagram as we discussed previously all unmanaged solutions co-exist in the same layer together. Each managed package however exists in its own layer on top of the built-in system solution. Users of the system ultimately see a calculated version of these layers referred to above as the "Active Customizations". The "Active Customization" view the user sees is a result of the merging of the layers and composition into a single view.

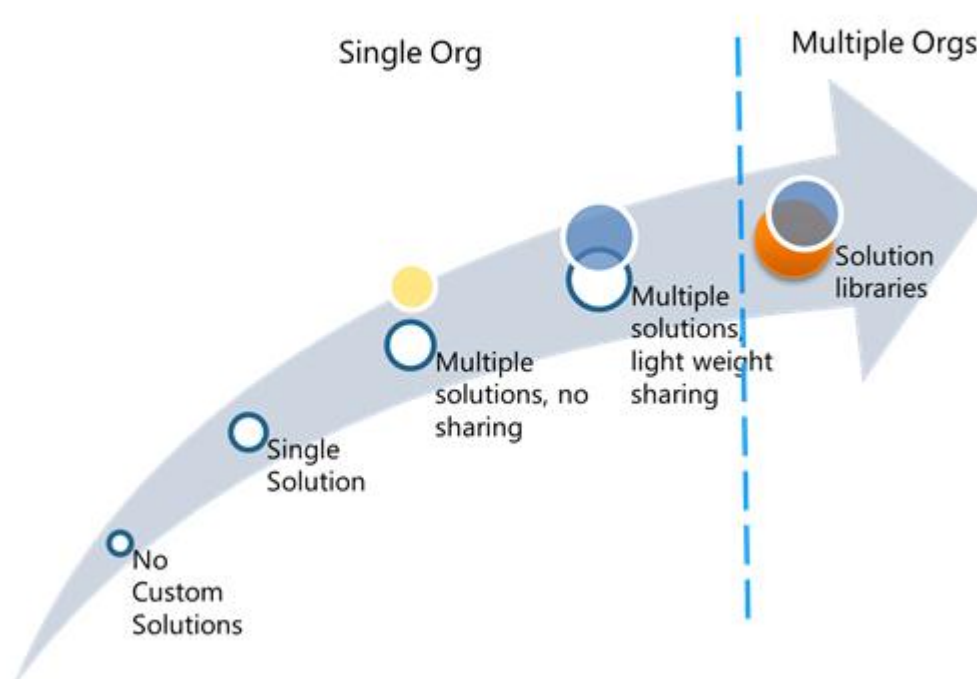
During the composition of the layers into a single view the system takes into consideration many factors. One key element of that is handling or avoiding conflicts that arise. In prior versions of CRM most elements only supported a last one wins resolution strategy. Many elements in Microsoft Dynamics CRM 2011 now support merging and co-existence. Examples of these elements are Ribbon, SiteMap and the Forms. An example of an element that still isn't merging enabled would be the display name for an entity. In that example there isn't a reasonable way to co-exist if ISV A wants to rename Account to Company, and ISV B wants to rename it to Brokerage Firm. A complete discussion of the merging rules can be found in the SDK under "Managed Solution Merging Rules".

Another area where the solution framework helps is maintaining customer driven customizations. In prior versions customers often made changes on top of ISV solutions only to find them lost after the next update from the ISV. The solution framework now has the ability to allow the customer to choose to maintain their customizations or overwrite them at the time they install an updated solution from the ISV.

For customers upgrading to Microsoft Dynamics CRM 2011 all prior customizations will be included as part of the unmanaged default solution. ISVs wanting to convert their solution to leverage the managed solution capabilities will have the option to install their managed solution on top of the upgraded prior components to convert their customers into a managed deployment.

Solutions are now a key aspect of defining the overall architecture of an Extended CRM application. There is no single right answer to choosing the sizing or complexity of the solution makeup of a product. ISVs should spend some time upfront in their planning stages deciding the structure of their solutions. The following provides some basic comparison of some of the different possible approaches.

Granularity = Flexibility = Complexity



While solutions are a powerful feature, ISVs should not look to them for providing digital rights management or other intellectual property protection for their product. ISV developers however could use the capabilities of the solution framework combined with plug-ins or other customizations to implement their own licensing mechanisms.

As you read through the rest of this paper, it's important to understand that the solution framework is enabled in almost all aspects of the new features. For example, when creating workflows, creating reports or customizing forms they all work with the solution framework to provide for a consistent customization experience.

Modeling Business Data

Dynamics CRM models business data using a mixture of built-in entities and custom entities. Entities can be thought of as similar to a database table. Each entity has attributes that are similar to columns on a database table. Relationships are created to connect the entities together to more fully describe the business connections that exist. Entities are the key to CRM's reusability and the built-in entities & modules can be re-purposed or not used from the system for building Extended CRM applications.

For each entity including the custom ones, the following is automatically enabled:

- Standard forms, views and access via the framework API
- Role based security - enforced via the UI and the API
- Activities - the ability to initiate and track a standard built-in set of activities (e.g. Tasks, E-mails, Phone Calls, etc.).
- De-Duplication - Flexible rules can be defined that identify & notify users of duplicate records
- Notes and Attachments - Simple notes and file attachments can be stored associated with the entity data records.
- Offline Support - entities can be enabled for offline viewing and use. Changes to the data are automatically synchronized back to the server.
- Charting, Dashboards and Reporting both for users via a wizard and for developers using standard tools
- Processes – Workflows background process automation, Dialogs for interactive user process automation

Think of these as part of the ripple effect of creating a new entity in Dynamics CRM 2011- while similar to a database table in concept, a CRM entity provides a good deal of functionality out of the box to speed an ISV towards shipping their application. The list goes on to include many other features but the point is; when an entity is created it automatically benefits from a number of capabilities provided by the platform.

Dynamics CRM 2011 introduces some new enhancements to how business applications can be modeled on the platform. The following are some of the key enhancements to this area:

- **Teams** – can now be owner of data records and have Security Roles
- **Custom Activities** – extending the built-in activities such as Phone, Task etc. to allow you add your own e.g. Instant Message, Building Showing or whatever custom activities are tracked in your application
- **Connections** - allowing ad-hoc dynamic relationships to be identified across data
- **Queues** – changing to User or Team Ownership and working with any entity
- **Lists** – support for dynamic lists membership
- **Goals** – Allows for setting, tracking and reporting on business goals.

As organizational structures become more nimble and dynamic, the "Teams" feature of Dynamics CRM 2011 becomes more important. Prior to Dynamics CRM 2011 teams could be granted access to data via sharing but could not directly own data records. Data records could only be owned by a single user or at the organization level. In reality many businesses have teams that share the ownership of the data- in a big real estate deal a team of sales people may be responsible for marketing a property. Dynamics CRM 2011 will not only allow security to be set at the team level but also enables team ownership.

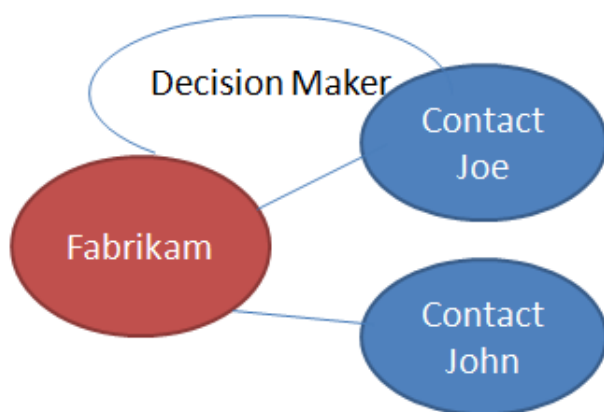
Custom Activities

Activities allow the tracking of interactions- typically interactions between people or between the system and people. Built-in activities like Tasks, E-mails and Phone Calls cover most common scenarios but Dynamics CRM 2011 now allows creation of custom activities that participate in the same way as the built-in ones. So for example it would be possible to create a custom activity to represent SMS text messages sent to and from a person's mobile phone. Creating a custom activity is just like creating a regular custom entity only it is marked as a custom activity at creation time.

Connections

Relationships between data can be modeled using the standard Dynamics CRM 2011 relationships (1-many, many-1, many-many) at the entity level. These types of relationships, while appropriate for some data, do not work for relationships that are more dynamic and ad-hoc. They also don't provide a way to track any associated attributes like the roles of the parties in the relationship. Relationship roles are another key aspect of modeling real world business applications. They provide the in-context connection between data records. Prior versions of Dynamics CRM had the Customer and Opportunity Relationship features that provided this for a small set of entities but it wasn't a flexible solution that could be extended. Dynamics CRM 2011 introduces the concept of Connections. Connections are flexible ad-hoc associations between records. The meaning of the Connection is expressed through a common set of attributes, most typically "role".

The following image illustrates a simplistic connection between a company, Fabrikam, and two contacts at the company, Joe and John. A Connection is used to represent the Decision Maker role that Joe has with Fabrikam.



The role can apply in different ways to allow Connections to enable a number of different scenarios. The following highlights how the role applies to connections:

- Applies to both parties
 - e.g. Friend, Golfing Partner, Spouse
- Applies to one party - the other party is either implied or unnecessary
 - e.g. Stakeholder, Influencer, Champion, Decision Maker
- Reciprocal - both parties described with a role
 - e.g. Employee - Employer, Father - Son

Connections can be used with just about any entity to create connections that span people, process and data. ISVs can use connections as part of their product offerings or create utilities that help visualize or otherwise work with connections at a more generic level. Between the built-in entity relationships and the new Connections, the platform now has support for both static design time and dynamic data/user driven relationship management.

Field Level Security

The need to secure data on a record can influence the data model and other aspects of an application's architecture. In prior versions of CRM users either had access to all fields on a record or none. In applications where there was a need to secure individual fields it often resulted in altering the data model to accommodate. Microsoft Dynamics CRM 2011 now includes support for Field Level Security. Once enabled, Field Security Profiles can be created that establish who (users or teams) are able to have access to the contents of the field. The Field Level Security is in force regardless if the user is interacting with the user interface, web services or other supported techniques for accessing the data.

Note: Field Level Security can be enabled for custom entities and fields however; currently you are not able to enable it on the built-in entities. Developers should look to prior version techniques for these fields for the time being.

Data Auditing

Tracking changes to a record's data is a common application requirement. In prior versions of CRM this typically required developers to implement the auditing using plug-ins or other custom code. Microsoft Dynamics CRM 2011 now has a built-in auditing feature. Auditing is enabled globally for the organization, then at the entity and individual field level. The following is an example of the output when Auditing is enabled on an entity.

The screenshot shows the 'Audit History' tab for a 'Contact' record named 'John Doe'. The left sidebar shows the 'Audit History' tab selected under the 'Related' section. The main area displays a table of changes with columns: Changed Date, Changed By, Action, Changed Field, Old Value, and New Value. A red arrow points to the 'Filter on: All Fields' dropdown, labeled 'Filter to sort by Fields'. Another red arrow points to the 'Audit History' tab in the sidebar, labeled 'Easily Discoverable Tab in Records Form'. A third red arrow points to the table, labeled 'Historical Trail'.

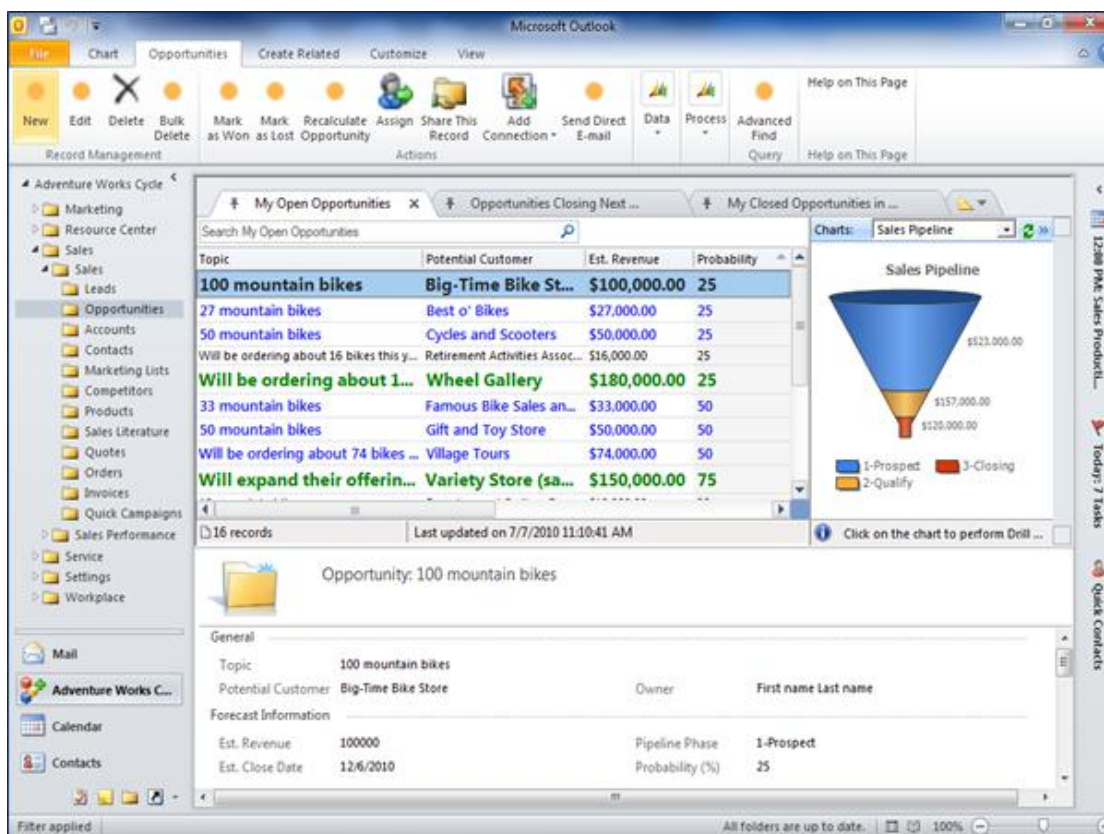
Changed Date	Changed By	Action	Changed Field	Old Value	New Value
7/24/2010 7:01 AM	Chris Admin	Update	Address 1: Name	11215, Research Blvd	11219, Oak St
7/24/2010 7:00 AM	Chris Admin	Update	Address 1: ZIP/Postal ...	78727	78734
7/24/2010 6:59 AM	Chris Admin	Create	Marital Status		Married
			Spouse/Partner Name		Betty Doe
			Territory		Default Value
			Address 2: Freight Ter...		Default Value
			Has Children		Default Value
			Address 2: Shipping ...		Default Value
			Is private		
			Do not allow Bulk E...		Allow
			Send Marketing Mate...		Send
			E-mail		jdoe@hotmail
			Education		Default Value
			Full Name		John Doe
			Address 1: ZIP/Postal ...		78727

Because of the positive impact of many of these data modeling and data management oriented features developers should be open to re-thinking how they might have solved the problem in the past. For example, in the past you might have repurposed a built-in activity such as Fax to represent an SMS message. Now in CRM 2011 you would probably be better creating and using a custom activity. As you start new solutions or make enhancements to existing solutions take a minute to reflect on how these new features might apply.

Building a Better User Experience

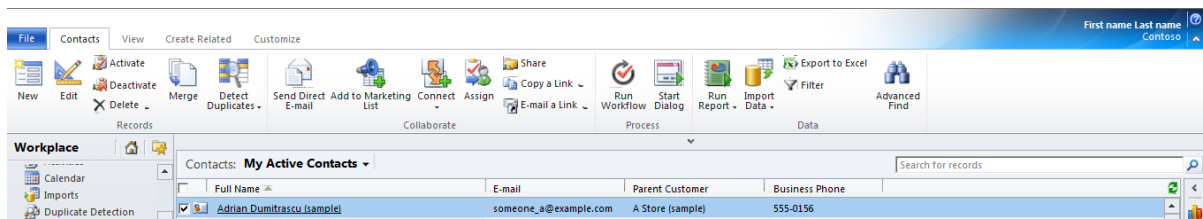
A major goal of Dynamics CRM 2011 was improving the user experience. This starts with fewer clicks to accomplish a task, and ends with a refreshed look at how CRM should be used within Microsoft Outlook. Microsoft CRM 2011 provides out of the box user interfaces via Internet Explorer, Microsoft Outlook and on the go with the Mobile Express interface. Each of these experiences can be customized using the built-in form editors and other customization techniques.

Microsoft Outlook is a key way users can interact with CRM. In prior versions, the CRM functionality was surfaced using web pages in Outlook. Microsoft Dynamics CRM 2011 now surfaces the CRM data using the standard Outlook controls when the user works with CRM data inside Outlook. By using the same control users get a familiar experience working with CRM inside of Outlook. This also enables features like personalization to tailor the grids as well as other common Outlook features like categories and follow-ups which act just like they do with other Outlook data. The following is an example of the CRM data surfaced within Outlook.



Microsoft Dynamics CRM 2011 now also sports the latest Microsoft Office style user interface that gives users a familiar navigation experience. This included things like introducing the Ribbon navigation to the Dynamics CRM 2011 clients.

The following is an example of the Dynamics CRM 2011 ribbon.



The ribbon allows ISVs more opportunities to customize navigation and command actions at the application and entity level. Included is the ability to add, hide and override commands. The ribbon can be customized to include new tabs, groups and controls. Several control types are supported such as Buttons, ComboBox, CheckBox, DropDown and more. Each element's display and enablement can be controlled by DisplayRules and EnableRules. These rules allow for business rules driven by a variety of triggers such as based on field value, entity privileges and existence of relationships to name a few. The SDK has full details on all the available control and rules that can be used. Dynamics CRM 2011 will start the process of retiring the current ISV.config file that was used for UI customization in previous versions of CRM. For example, the Toolbar and Menu definitions have now moved to be part of the ribbon configuration. Additionally, since the ribbon will participate in the Solutions framework it will improve the ability for multiple ISV applications to co-exist in the same end user deployment.

At the same time, a lot of effort was also focused on direct user feedback such as making the user experience less "clicky" than prior versions. This comes into play at a number of places in the new Dynamics CRM client interfaces. For example, instead of using tabs, forms now allow for longer pages with simple navigational links. This removed the limit on number of tabs that previously existed. Sub-grids of related data shown inline on a form are now possible out of the box rather than having to resort to the use of iFrame hosted views.

Enhancements were also made for those who customize CRM with developer productivity as a key focus. Customizations can be made directly from entity forms using the "Customize" tab. Form editing is simpler because you can now drag and drop fields onto the form designer surface and on the fly custom field creation from the form designer surface as well. If you make a mistake, undo and redo is now possible. Form layout is now also more flexible allowing things like side-by-side sections and up to four fields across the page.

Web Resources is a new feature of Dynamics CRM 2011 to provide a place to store and render client side content. Web Resources are automatically deployed to offline clients to provide a consistent experience regardless of a user's on/off network status. The following are some of the items that can be stored as web resources:

- Applications - Silverlight®
- Image - PNG, JPG, GIF formats
- Text based - HTML, XML, XSL, CSS, JS

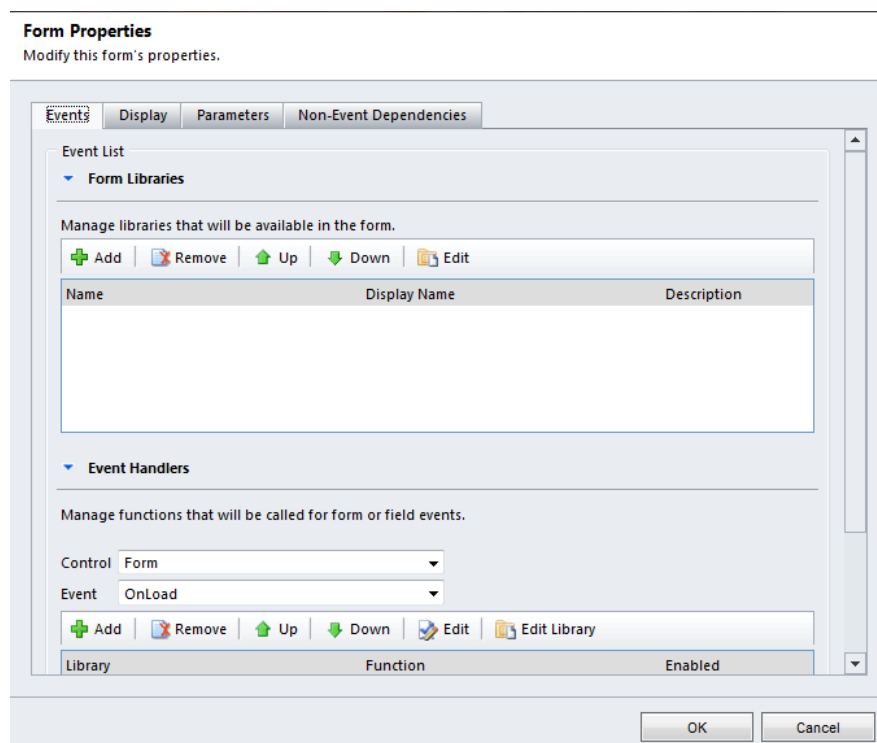
Web Resources, once defined, provide direct URL access. Using the Form Editor (and via formXML) you can easily insert HTML content via an iFrame. Both images and Silverlight controls can be inserted in-line without the need for an iFrame. Unlike server hosted pages, HTML and Silverlight content defined as a Web Resource is deployed as part of a solution and available for use by the user when they are off-line.

Note: It is important to be aware that when using CRM Online there is no provision for hosting ASP.NET pages directly in CRM Online. You can however use web resources as your canvas (HTML or Silverlight) and plug-ins as the back-end services. ASP.NET content can also be hosted in Windows Azure and use federated identity to allow single sign on to the ASP.NET content.

Web Resources can host a variety of content with different levels of complexity. These resources can be called upon in several areas of the xRM solution. The ability to use and re-use these resources quickly increases the value of the ISV solution built upon the xRM Framework.

JavaScript Libraries are another concept that is new and builds on the Web Resource feature. JavaScript libraries are client side snippets of JavaScript that can be reused across multiple Forms, Ribbons, etc. Form events will be able to take advantage of the JavaScript libraries by providing special directives to include and consume them. Further, form events will become more of a pipeline allowing execution of multiple event handlers defined from the JavaScript libraries.

The following image shows an example of registering multiple JavaScript libraries with a form.



The client scripting model has been further enhanced for Dynamics CRM 2011. The new scripting model provides developers more flexibility when working with the form controls and the data associated with them. Developers are no longer tightly coupled to working with DHTML controls directly from script. Several of the common unsupported scripting needs from the past have now been accommodated in a supported way within the new client scripting model. For example, *getUserRoles* method gives a quick way to get all the roles of a user. *AddCustomView* method on a Lookup control allows adding a custom filter. Backward compatibility has been maintained for developers using the prior client scripting model in a supported fashion. You can find complete details in the SDK under the Client-Side Programming Reference topic.


Role Based UI

Another common request for improving the overall application experience is allowing users in different roles to see different content on the standard forms. For example, a Leasing Agent and the Property Manager in a real estate business need different information about a real estate property. In prior versions there was only a single main form without the flexibility to show and hide based on the security model- there were of course a variety of stop-gap solutions implemented by creative developers. New to Dynamics CRM 2011 is the ability to have role based pages without the creative measures of before. These pages extend the existing Form/Page model to support multiple pages. Standard user roles are used to determine the page content that a specific user will see based on their role. This means that all entities are automatically enabled for this feature.

Additional pages can be created either by saving a copy of the existing main form or by creating a new one. The following is an example of an entity having multiple forms.

Name	Form Type ▲	State
<u>Information</u>	Main	Unmanaged
Sales Agent View	Main	Unmanaged
Property Manager View	Main	Unmanaged
Information	Mobile	Unmanaged

Forms can now be assigned to roles using a simple dialog that will allow you to indicate if a form should be shown to all users or just those in specific CRM roles.



Assign Security Roles: Property Manager View -- Webpage

Assign Security Roles: Property Manager View
Select the security roles for which this form will be displayed.

☐ Display to everyone

☒ Display only to these selected security roles

<input type="checkbox"/>	Name
<input type="checkbox"/>	Delegate
<input type="checkbox"/>	Leasing Agent
<input type="checkbox"/>	Marketing Manager
<input type="checkbox"/>	Marketing Professional
<input checked="" type="checkbox"/>	Property Manager
<input type="checkbox"/>	Sales Manager

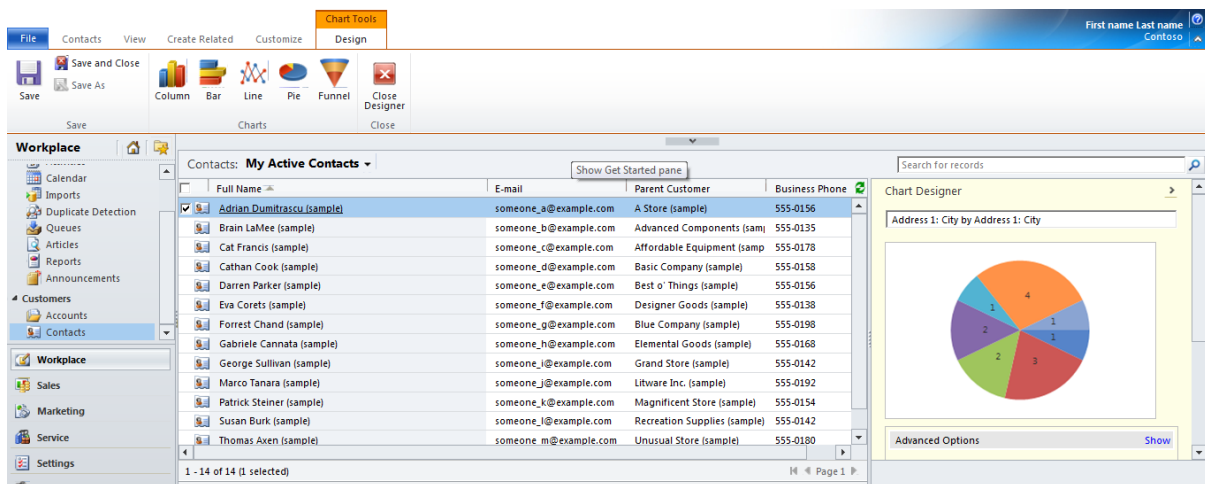
Users accessing an entity record that has multiple forms can choose to switch between any available forms that are available to their roles. Developers can also programmatically switch the form that is displayed using client side scripting. The current support for determining the page to use is driven by roles and not data driven. Developers can however use the scripting capabilities to implement data driven rules if needed. In general though, using the client scripting to hide and show tabs or sections on existing forms will often perform better and not cause a complete page refresh.

Visualizations – Charts, Dashboards and Reports

The data input process of an application is important, but often it is the ability for a business to analyze the data and visualize it in meaningful ways that make the difference. Microsoft Dynamics CRM 2011 enables this on built-in data and custom entities created for the application. Using Charts, Dashboards and Reports, combined or individually, users and developers can build visualizations to tell a more complete business story. The data can be displayed at a summary level and then allow users to drill down and take action on individual data rows. Users are empowered to use the built-in tools to create the visualizations starting from scratch or from templates provided by the ISV. Developers can pick up from what users started, or also create new visualizations from scratch. Developers can use the additional tooling provided by Microsoft Business Information Development Studio (BIDS) which provides them a familiar Microsoft Visual Studio editing experience.

Charts

Charts and Dashboards provide users a higher level perspective of their business data. One or more charts can be defined and associated with an entity. Those charts become available to the user under the view tab and appear alongside a grid of entity data. Users then get an in-context visual representation of the grid data. Microsoft Dynamics CRM 2011 provides a built-in Chart designer that allows users to build all the common charts in the grid area. The following is an example showing a grid of data with a simple Pie Chart.

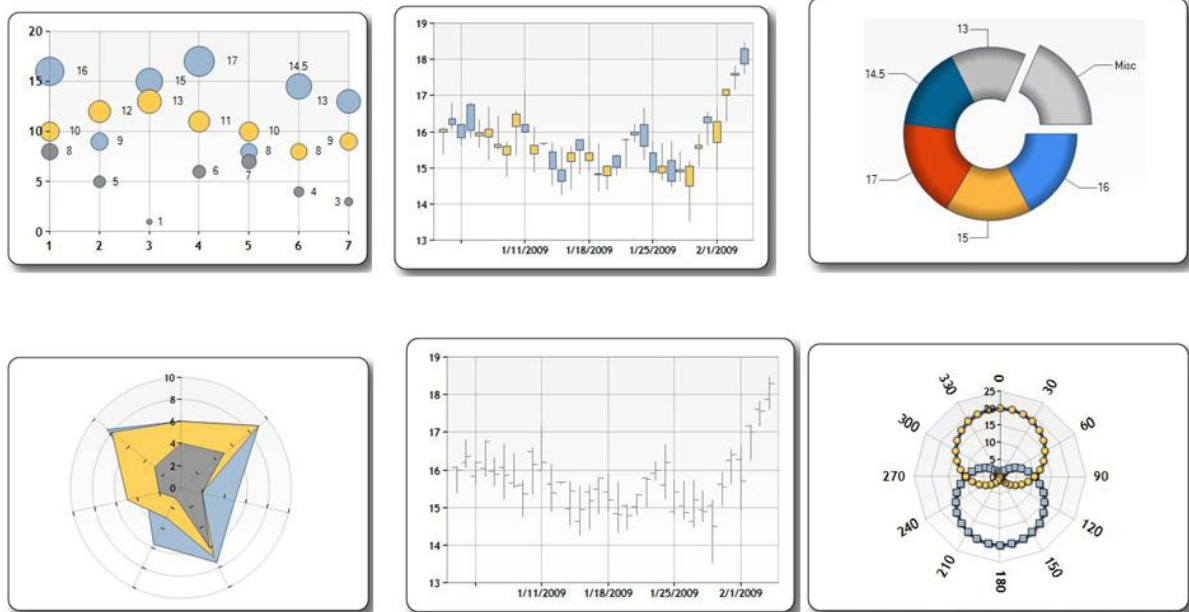


Charts support multiple levels of drill down allowing users to drill down into the different segments of data. As the user drills down the view shown in the grid is updated and they can also select different chart types that are appropriate for the lower level of data.

New chart visualizations can be easily created by end-users. From the View Ribbon they can start the Chart Wizard to customize and select the type of chart to display. Charts can be imported & exported and can also be included in a solutions package. Standard security applies allowing charts to be shared with other users just like you would share a report or be shown to the organization as a whole when made as part of a solution.

Developers can further customize the exported chart to perform formatting and other changes that are more advanced than those supported in the wizard. This includes using the full set of chart styles supported by the ASP.Net charting controls library.

The following is an example of a few of the ASP.NET charting controls that are not included but can be enabled by developers editing the exported Chart.

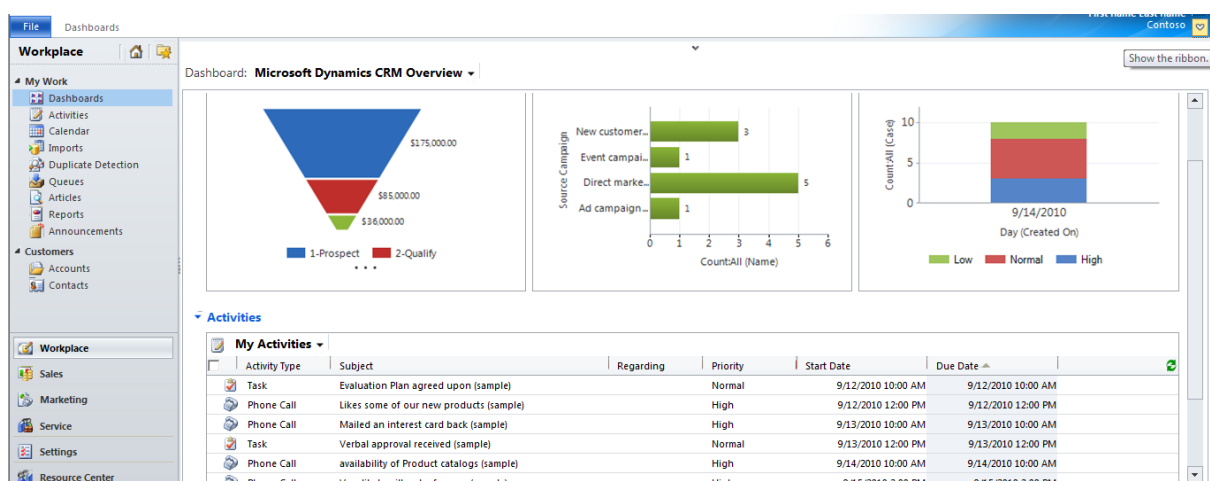


Using this approach gives developers access to 35 different ASP.NET charting controls.

Expect to see charts being used a lot more frequently than reports because they can easily be displayed in-line with the grids of entity data.

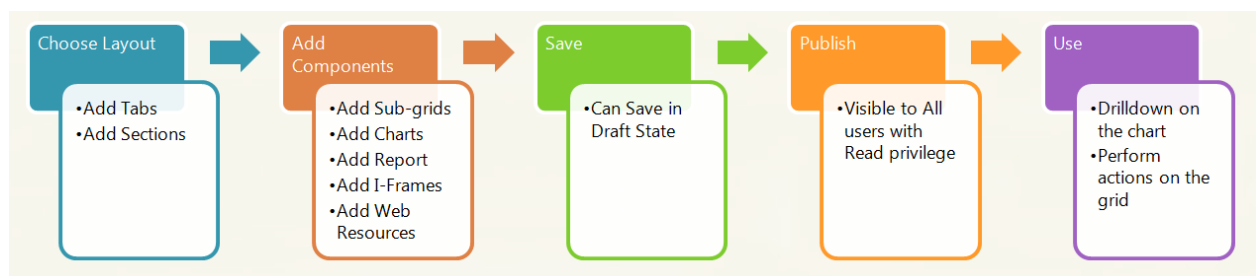
Dashboards

Dashboards provide a simple way for users to see a variety of views of data from an application in a single place. Typically the view saves the user time from going to several different places to see key metrics. Dashboards can contain data from several different entities, showing saved charts, views and web resources all in one screen. The following is an example of a system dashboard.



Dashboards can either be system dashboards where all users see them or user specific dashboards where they are tailored by a user to their particular need. Dashboards are based on the Form Storage Model (FormXml) and can be included as part of a Solution package. This means that ISVs can include one or more dashboards as part of their products. These custom dashboards can contain data displayed using the standard xRM Framework visualization charts, grid views of data records and web resources such as Silverlight and HTML. Each dashboard can contain displays from a variety of sources and still be shown together in a single user view. In addition, the dashboard items can bring their native functionality with them; charts still have drilldown ability and grids can still be sorted or searched.

The following highlights the development process for a dashboard. You can also see in the second step some of the different types of components that can be included as part of a dashboard.



Reporting

Reporting in Microsoft Dynamics CRM 2011 is handled by Microsoft SQL Server Reporting Services. Users are able create, modify and run reports from within the user interface. Developers can use the standard Reporting Services tools to build and customize reports. In the past, these developer created reports were not able to be deployed to CRM Online. Microsoft Dynamics CRM 2011 introduces a new extension to Microsoft Business Information Development Studio that allows the FetchXML syntax for query definition instead of SQL. Reports developed using the FetchXML syntax for the query can now be deployed to CRM Online as well as on-premise and partner hosted deployments.

The following is an example of the Query Designer within Microsoft Business Information Development Studio (BIDS) showing a preview created from FetchXml.

Query Designer

Command type: Text

```

<fetch version="1.0" output-format="xml-platform" mapping="logical" distinct="false">
  <entity name="contact">
    <attribute name="fullname" />
  </entity>
</fetch>
  
```

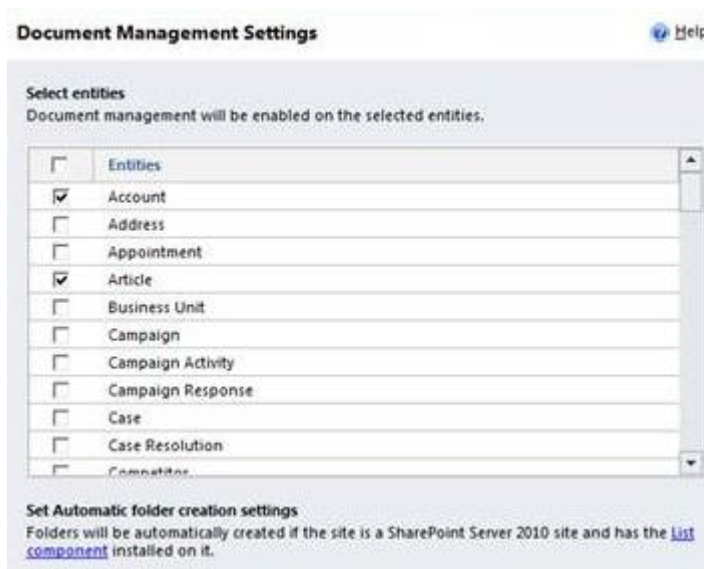
fullname	contoso_type	contoso_typeV...	parentcustome...	parentcustome...	parentcustome...	contactid
Adrian Dumitr...	Fitness Center ...	100000000	<NULL>	<NULL>	<NULL>	c617be6b-10c2.
Alan Brewer	Fitness Center ...	100000000	<NULL>	<NULL>	<NULL>	8417be6b-10c2.
Alan Waxman	Fitness Center ...	100000000	<NULL>	<NULL>	<NULL>	8c18be6b-10c2.
Alex Hankin	Fitness Center ...	100000000	<NULL>	<NULL>	<NULL>	ea17be6b-10c2.
Alex Litton	Fitness Center ...	100000000	<NULL>	<NULL>	<NULL>	2218be6b-10c2.
Alice Ciccu	Fitness Center ...	100000000	<NULL>	<NULL>	<NULL>	a217be6b-10c2.

Developers can then use all the formatting and preview features of BIDS to customize reports they are creating. Once completed the reports are uploaded to CRM and are available for users to run from within the normal CRM user interfaces.

Integrating with SharePoint®

Microsoft SharePoint has rich document management and collaboration features. In fact, many CRM implementations include integrations to Microsoft SharePoint. Prior to Microsoft Dynamics CRM 2011 this integration was a manual effort. Dynamics CRM 2011 now has built-in integration with SharePoint for document management. From CRM it is now possible to associate Microsoft SharePoint Document locations to a CRM record. Once associated, users are able to collaborate on documents in the context of CRM records. For example, multiple leasing agents could collaborate on a leasing proposal prior to sending it out to a client. They would find the document by looking up the property record in CRM and then navigating to the documents section. There they would find the SharePoint document library surfaced listing the leasing document. Using standard SharePoint collaboration techniques the leasing agents would collaborate on the document. SharePoint integration is enabled when Microsoft SharePoint 2007 or Microsoft SharePoint 2010 are installed.

Microsoft SharePoint integration can be enabled for both built-in and custom entities using the following dialog.



Once SharePoint is enabled for an entity and SharePoint also has the CRM List components installed, folders are automatically created as users work with the records in CRM.

The built-in integration of SharePoint and Dynamics CRM 2011 is limited to document management in this release but other features of SharePoint can be leveraged using the SharePoint SDK.

Dynamics CRM 2011 Programmability Overview

The rest of the paper will take a look at some of the developer focused capabilities of Dynamics CRM 2011. These are the lower level code focused techniques that can be used to accomplish the more advanced customizations not possible with the customization screens in the client interface. In fact, just about everything that you can do in the client interface can be also done by a developer using the framework API.

Developers working with the xRM Application Framework will use Microsoft Visual Studio 2010 as their primary tool. Using Visual Studio developers can connect to the xRM services, develop extensions and embed custom logic and user interfaces using .NET and ASP.NET, WPF, or Silverlight as appropriate for the type of applications.

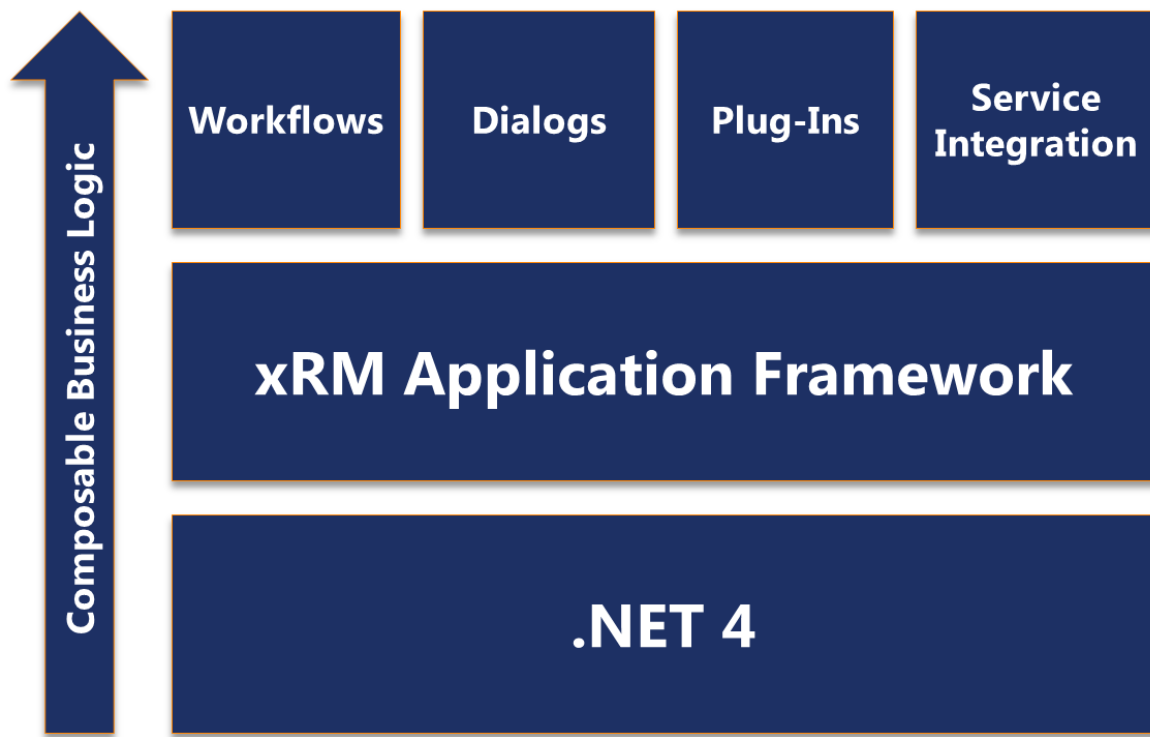
Developers do have additional hooks into the platform processing using the unified event pipeline. The event pipeline comes into play anytime data is created or modified and it creates an event pipeline that handles processing the system request. As the request is processed it creates several opportunities for a developer to register to have custom code executed. This custom code can be implemented as a framework plug-in or a workflow. Plug-ins can be processed synchronously or asynchronously depending on how they are registered while workflows always run asynchronously. Plug-ins, once registered, execute silently behind the scenes and are typically transparent to the user. Workflows on the other hand are visible to the user and can be monitored from the workflow section on each entity form. Both workflows and plug-ins can be registered to be triggered based on specific type of events like Create, Update or Delete. Workflows can additionally be manually invoked on demand by a user. Both plug-ins and workflows can be as simple as performing a calculation or more complex and use the APIs we will talk about later in the paper.

Dynamics CRM 2011 introduces a lot of change in the area of developer programmability. In fact there are major changes to working with data and services, plug-ins and workflows. Most important though is that prior investments are maintained because Dynamics CRM 2011 provides backward compatibility in each of those areas. No change or very minor changes are required to prior developed code to migrate to Dynamics CRM 2011. In the cases where you want to leverage some of the new features is where you might have to modify your code to work with the new versions features.

Business Processes in Dynamics CRM 2011

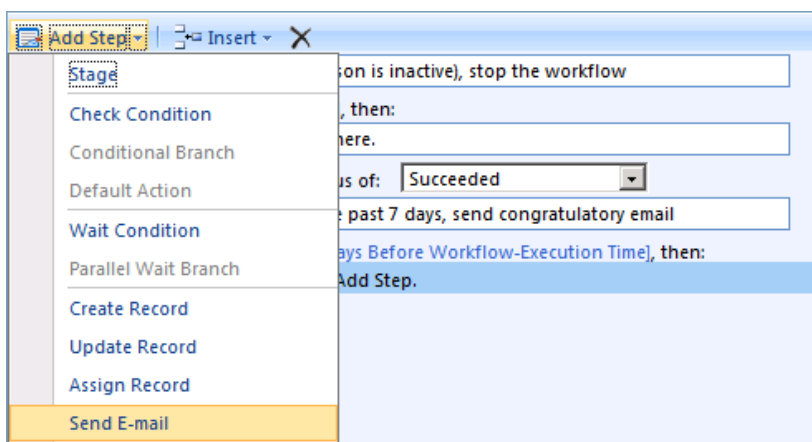
The overall options for automating business processes continue to evolve in Microsoft Dynamics CRM 2011. Workflows will now be found under the header Processes and you will also find Dialogs there as well. Dialogs are new to Dynamics CRM 2011 offering a way to provide guided processes that are like workflows but interactive with users.

The following diagram shows the different options for automating business processes in Dynamics CRM 2011.

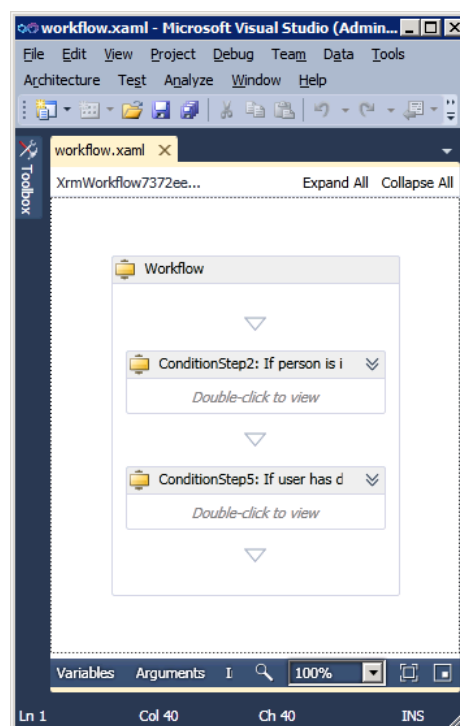
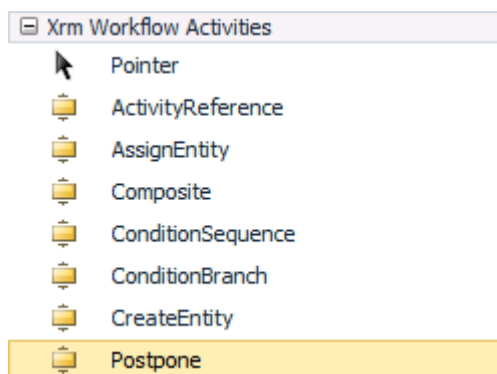


Workflows

Workflows can be built by users using the web interface or by developers using Visual Studio. It is also possible for a workflow to start out being built by a user and then extracted by the developer for further customization. This is possible because Windows Workflow Foundation 4 is now used as the workflow engine. Windows Workflow Foundation 4 now uses XAML (a declarative XML based specification) to specify the workflow. Users can edit the workflows from within the web client using the workflow editor as you see in the following image.



The workflow can be extracted and edited from within Visual Studio using the built-in workflow designer and the xRM specific workflow activities as you can see in the following example.



Note: Support for modification of workflows in Visual Studio and custom activities are not currently supported by the security sandboxing in CRM Online. Developers needing this type of support should look to the command pattern with plug-ins or other alternate solutions for the time being.

Dialogs

Dialogs are new to Dynamics CRM 2011 offering a way to provide guided processes that are like workflows but interactive with users. They share much of the same underlying infrastructure as workflows but add an interactive capability. These guided processes fit the obvious need of call center scripting, but also can be used to automate other common interactive business processes. For example, in a real estate management solution you could use the dialog feature to script a basic creation of a property appraisal request. Prompting the user through a series of questions tailored based on the property type.

Dialogs processes are created using either the SDK or the web client editor. The editor is very similar to the Workflow editor in how it builds the dialog. It uses the same type of techniques for things like adding steps, setting data to fields on the entity records. It also has Dialog specific features like the ability to run a query and use that as part of the process as well as the idea of a page the user interacts with.

The following is an example of the Dialog editor and setting up a page that will prompt the user to choose what the purpose of the appraisal is.

The screenshot shows the 'Define Prompt and Response -- Webpage Dialog' editor. The left pane shows a tree structure with 'Input Arguments', 'Variables', and 'Steps'. The 'Steps' pane is expanded, showing a step named 'Page: Purpose of appraisal'. The 'Prompt and Response' section is visible, showing a 'Prompt' text: 'What is the purpose of the appraisal of property {Name(Commercial Property)} ?'. The 'Response' section is empty. The 'Prompt Text' field is also visible, containing the same text: 'What is the purpose of the appraisal of property {Name(Commercial Property)} ?'.

Once the Dialog is built, just like a workflow it is activated and then becomes available for use by users. Users will find the available Dialogs for an entity by clicking on the Start Dialog button from the Ribbon. It is also possible to have a button invoke a specific dialog on demand without showing the user a list. For example you could add a "Request Appraisal" button to the ribbon to make it easier for the user to perform common tasks. Using the built-in Start Dialog button the following is an example of what the user would see.

The screenshot shows the 'Look Up Record -- Webpage Dialog' box. The title is 'Look Up Record'. The instructions are: 'Enter your search criteria and click Search to find matching records. Filter your results and view different columns of data by using the View options. Then, select the record you want and click OK.' The 'Look for' dropdown is set to 'Process'. The 'View' dropdown is set to 'On Demand Dialogs'. The 'Search' field is empty. The 'Show Only My Records' checkbox is unchecked. Below the search fields is a table of results:

	Process Name	Created On	Modified On	Status	On
<input checked="" type="checkbox"/>	Property Appraisal Request	10/10/2010 1:37...	10/10/2010 1:40...	Activated	Fir

Once the user selected a Dialog to start the first page would be shown after the prior steps had processed.

As you can see in the following example the Page and Prompt that was built is now shown to the user for selection.

The screenshot shows a web browser window titled 'Property Appraisal Request - Windows Internet Explorer'. The page has a header 'Property Appraisal Request'. On the left, a sidebar titled 'Commercial Properties:' contains a list with 'Name' as a header and 'One Microsoft Way' as a selected item. The main content area contains a blue prompt box with the text 'What is the purpose of the appraisal of propertyOne Microsoft Way?'. Below the prompt are two radio button options: 'Property Sale' (which is selected) and 'General Property Valuation'.

Workflows and the Solution Framework

Workflows, Dialogs and Custom workflow activities are now supported by the Solution framework. Developers continue to create the custom activity in Visual Studio. The Plug-in Registration Tool is used to register the custom activity with CRM. The .NET assembly containing the custom activity is stored in the CRM organization database. From the Solution menus the custom activity can be associated with the solution that is being developed. Workflows and Dialogs can also be created in the context of a Solution or associated with one using the Add Existing feature. Once these components are associated with a solution any export and import of that solution will contain those components. When installed in a target system using the import feature the user can choose if they want to activate the components at that time or do it manually later.

CRM Workflow Upgrade

Workflows present an interesting challenge for migration from older versions due to the move to .NET 4 Windows Workflow Foundation. Existing workflows are upgraded as part of the standard upgrade process. This includes unpublished, published, running and completed instances. For any running workflow any persisted state must also be upgraded. Any .NET 3.x custom workflow activities are wrapped during the process in the .NET 4.0 "interop" activities allowing them to still run after the upgrade.

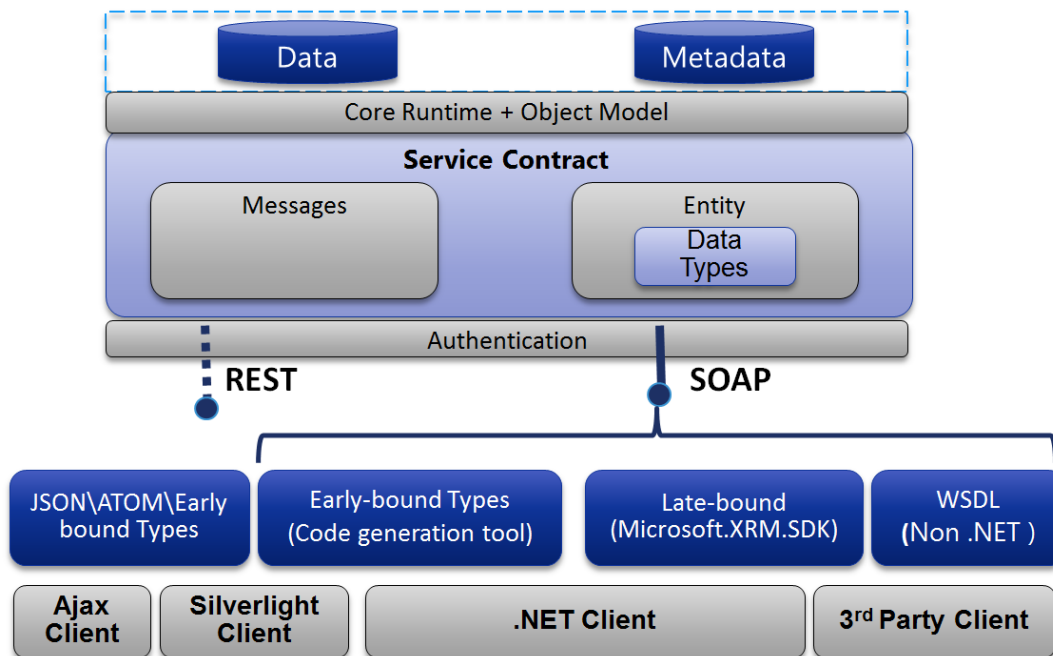
Workflow Assembly Versioning

In previous versions of CRM introducing an upgrade to a custom workflow activity assembly was challenging. In fact, a common workaround was to just create a new named control to avoid the challenges all together. Microsoft Dynamics CRM 2011 will now use the version number of the workflow activity assembly to control the versioning. Non-breaking changes will allow in-flight instances to now use the new assembly. For breaking changes, the old and the new assembly will be allowed to co-exist. Existing workflows will continue to use the old assembly.

Processes in CRM continue to take on more aspects of the business process automation in a solution. The addition of Dialogs fills a prior void of being able to automate the interactive processes. Having the Process components participate in the Solution framework also makes deploying solutions with business automation components easier.

Working with Data

Dynamics CRM 2011 introduces some new ways for developers to work with data. A WCF (Windows Communication Foundation) SOAP endpoint is available for .NET and other 3rd party clients to work with. Ajax (client scripting) and Silverlight clients can use a new REST endpoint that implements the OData (Open Data) protocol. This new interface provides the client extensions a standard supported way to work with the data without having to hand craft SOAP requests. The following diagram shows the high level data access architecture in Microsoft Dynamics CRM 2011.



WCF Endpoint

Dynamics CRM 2011 introduces a new streamlined WCF endpoint that takes the place of the prior web services. The WCF endpoint is a consolidated API that includes both the CRM Service and the Metadata Service API in prior versions. The WSDL (service contract definition) is static and is the same for any organization. Some of the key benefits of the new WCF endpoint are:

- Streamlined API with a focused set of methods
- Use of standard .NET Types in most cases
- Ability to take advantage of WCF capabilities like binary encoding
- Improved performance


The service definition is static and that means that all interactions with the API that are entity related will use a generic Entity class. The Entity class is similar to the DynamicEntity class from prior versions of Dynamics CRM. Developers can interact with the API using the Entity class in a late bound fashion (working directly with the attribute collection). This type of access is ideal where you don't know ahead of time the entities that your application will be working with. Using organization or solution specific entity classes allowed the developer to work with typed classes. They would get full Intellisense® in Visual Studio and the chances of a typo in an attribute name causing a runtime error was next to none. Dynamics CRM 2011 provides a utility (CrmSvcUtil) that uses the platform metadata to generate similar typed classes for each entity. These classes inherit from the generic Entity class and allow developers to work with the known properties.

Another significant change in the API is around the data types used for each of the properties. In the past Dynamics CRM used a set of custom classes to represent the core .NET data types. For example, to represent a Boolean value a `CrmBoolean` class was created. A significant reason for that was the lack of nullable support in the core types. Now that the .NET types support nullables it is no longer necessary for many of the properties to be Dynamics CRM specific. The new API now uses standard .NET nullable types for most of the data types.

Using the LINQ Provider

Dynamics CRM 2011 introduces a new Language-Integrated Query (LINQ) provider for working with data. This is in addition to and sits on top of the WCF interface to provide an additional layer of abstraction and ease of use. The `OrganizationServiceContext` class can be used to build applications that are similar to WCF Data Services `DataServiceContext`. Developers will find the syntax for building queries with the LINQ provider simpler than the using the **QueryExpression** or **FetchXML** style of queries. You can work with the new LINQ provider using both early and late bound data types. Using the LINQ syntax will be familiar to non-CRM developers because it is similar to what they use with traditional database applications. The following is a simple example of a query composed with the LINQ syntax.

```
var query = from acct in ctx.AccountSet
            where acct.Name.Contains("a")
            orderby acct.Name
            select acct;
```



OData and the REST Endpoint

Open Data Protocol (OData) is a web protocol for querying and updating data. OData is designed to make application data easily usable in a consistent way regardless of its source. The source can be a database, SharePoint, or Dynamics CRM! OData has **open** in the name as a way to add emphasis to the fact that it builds upon existing standards such as HTTP, Atom Publishing Protocol (AtomPub) and Java Script Object Notation (JSON). Use of these well-established standards makes the data published by OData publishers easily consumable.

OData is important for xRM developers for at least two reasons. First, Dynamics CRM 2011 is an OData publisher and provides a new end point that you can access the CRM data using the OData protocol. Second, xRM applications are often composite applications built from a number of different sources. OData is a unifying technology making many sources look similar by providing a familiar interface on top of their unique underlying structure. When building composite applications having many sources a familiar interface can speed the ability to integrate the sources into a combined solution.

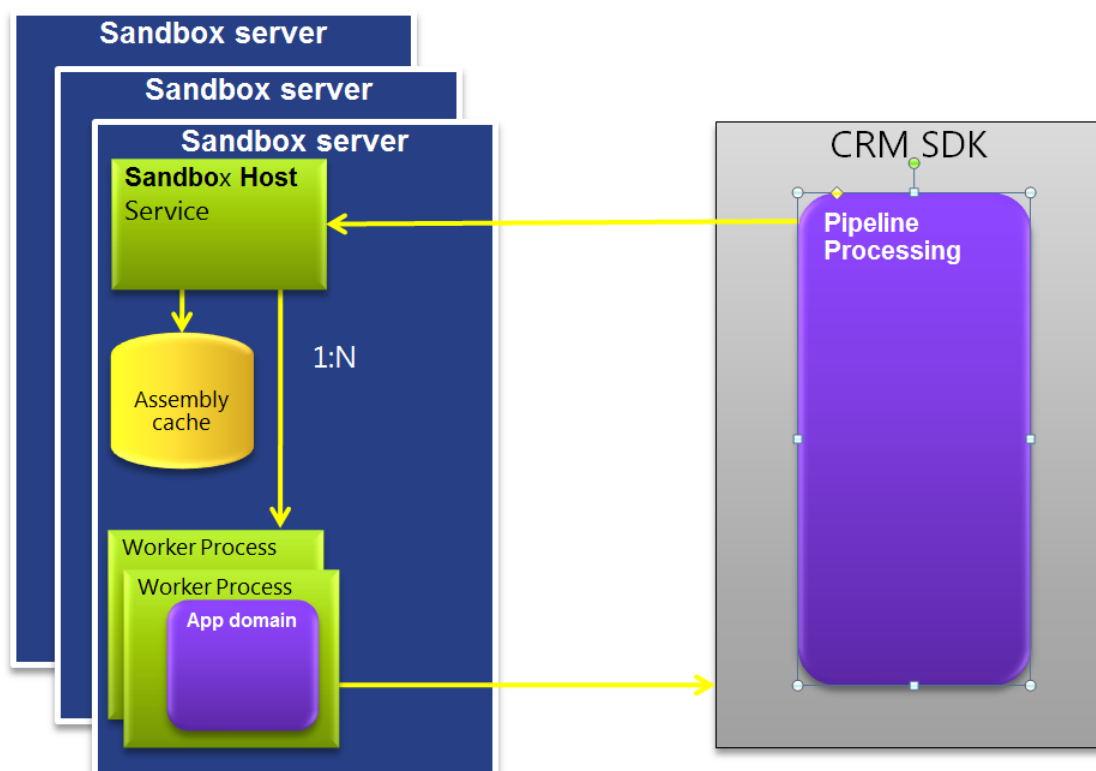
By making the CRM data available to client extensions using the OData protocol developers can use familiar LINQ style queries to work with the data. The requests interact with the CRM server using HTTP requests/responses. On the server side the requests pass through the WCF interface and all of the registered plug-ins just like any other data interaction request.

Plug Into the Sandbox

Plug-ins are an important part of the extensibility story for Dynamics CRM 2011. They allow an application to register for events and be invoked as part of the pipeline of processing a platform request. Prior to Dynamics CRM 2011 plug-ins were only able to work on-premises and in partner

hosted environments when allowed by the hosting provider. CRM 4.0 Online did not have the ability to deploy plug-ins. Dynamics CRM 2011 changes that by deploying a sandbox concept to allow plug-ins to be deployed in a reliable manner in on-premises, partner hosted and CRM Online.

Dynamics CRM 2011 plug-ins leverage the .NET CLR code access security to enforce a minimum set of privileges for the plug-in. Additionally, the process executing the plug-ins will use a low privilege account that has limited database access. A new separate server role will be used to isolate the process that runs the plug-in code. A separate process will be used to further isolate plug-ins for different organizations.



Plug-in logic can now participate in the transaction done for the framework operation. The work done by the plug-in will then be committed or rolled back depending on the outcome of other similar plug-ins and the framework operation. The WCF endpoint does not support Client side transactions; however this is one way to provide transactions across different requests.

The event pipeline for the plug-in was also streamlined to no longer have a concept of child / parent pipeline externalized to the developer.

A new tracing service is now available for plug-in developers to make diagnostics easier when errors happen during plug-in execution. These errors are available in an additional error log that can be reviewed by the ISV and then debugged.

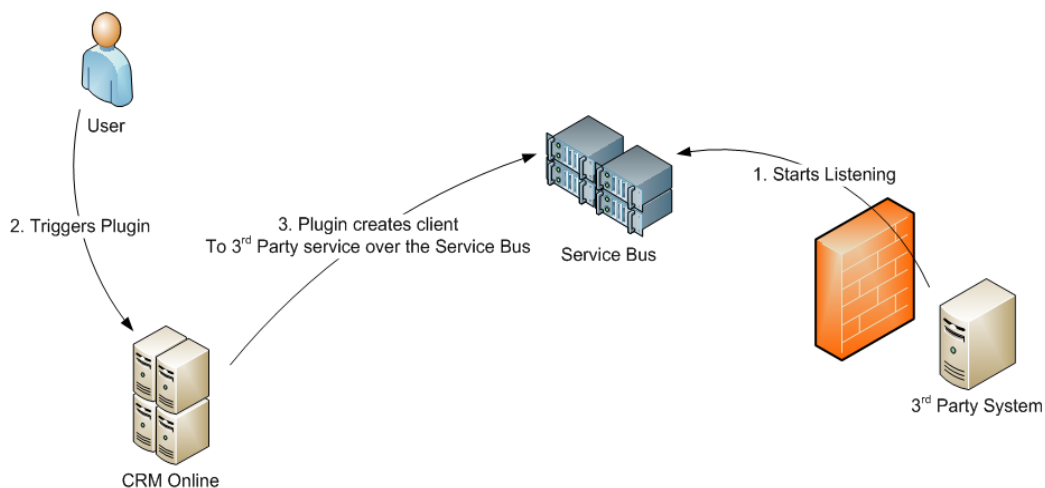
Plug-in monitoring is also a key part of ensuring reliability by watching the errors from plug-ins and disabling the plug-in as needed to ensure system stability. Plug-in monitoring will look for unexpected results and exceptions as well as excessive resource consumption. When excessive consumption is detected the system will disable the plug-in assembly to ensure system stability.

With these changes in Dynamics CRM 2011 plug-ins can now be deployed to the CRM Online environment. These plug-ins, while running in a secure Sandbox, can still reach out to external services via Web Service calls. For example, when a new real estate property is created a plug-in could be invoked that calls an address verification service to clean the address.

Event Pipeline and Windows Azure App Fabric Service Bus

Business processes can often cross the traditional boundaries of an organization. This can occur when sending data to partner companies or reaching out to access a 3rd party solution. As ISVs it might be desirable to have a service that you host that can be accessed by your customers outside their firewall. Dynamics CRM 2011 enables these types of scenarios by integrating with the Windows Azure Service Bus. The application does not have to pull for new messages; they are pushed across the service bus to the application based on event registration in CRM. Windows Azure AppFabric Service Bus also helps reduce the complexity of security between the two parties. Each message received will identify the organization it was sent from keeping the ISV from needing credentials to each organization it receives messages from. There is also no need to open firewall access to the ISV to the CRM server to allow direct access. This integration can be very useful when building applications for the cloud using CRM Online.

A new interface `IServiceEndpointPlugin` is available in Dynamics CRM 2011 for implementing an event listener. Azure AppFabric services can authenticate and route the request to the listener. The `IServiceEndpointPlugin` is similar to the `IPlugin` interface used by a traditional plug-in but is designed to work with the service bus. This remote plug-in is called by Dynamics CRM 2011 using the standard service bus APIs. An event registration is done to allow Dynamics CRM 2011 to know when to call out to the remote plug-in. The event registration identifies the event that is to be published to the service bus and the end point that should be hit. The events are posted asynchronously from Dynamics CRM 2011 to the service bus. In the event an error occurs reaching the service bus a retry is attempted. The .NET queuing feature of Windows Azure can also be used to stage messages. Using this support will be the pattern for applications that have a need to call out to other services. Support is provided for building one-way, two-way, queued and REST listeners. The following is a simple illustration of how an event might be published from CRM Online to a third party service that is hosted in Windows Azure.



Moving to Claims Based Security

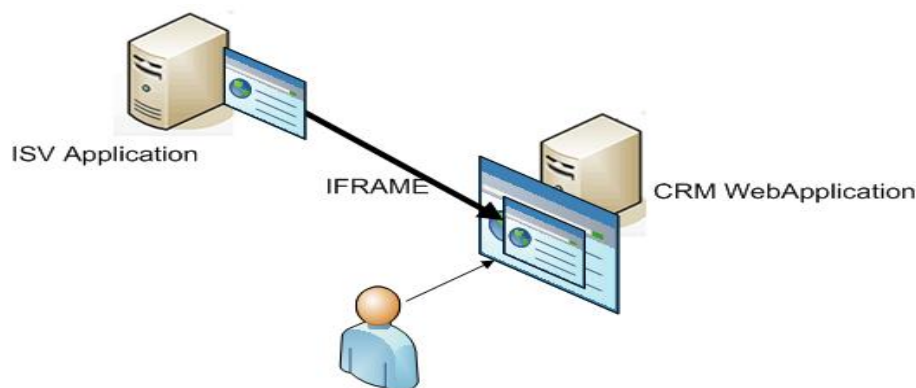
The concept of one size fits all security doesn't work very well to support the security needs of modern applications. The different hosting scenarios from on-premises, partner hosted and CRM Online also create additional security challenges. Developers don't have the time to become security experts in all the different possible authentication technologies. This problem is not unique to Dynamics CRM. In fact it's a challenge across Microsoft's platform tools and other vendors as well. Microsoft has created Windows Identify Foundation (WIF) formerly known as "Geneva" Framework to address this challenge. WIF is a framework for implementing claims based authentication that is utilized by Dynamics CRM 2011. Claims based authentication allows a user to present their identity to an application as a set of claims. For example one of the claims could be a user name. Dynamics CRM 2011 then takes that claim and presents it to the configured trusted external identity system or a Security Token Service (STS). The STS is then responsible for the actual authentication of the claims and not Dynamics CRM 2011. By moving the actual heavy lifting outside of Dynamics CRM 2011 to WIF it becomes easier to implement multiple authentication strategies. The following are some of the scenarios enabled by moving to claims based authentication:

- Support for existing AD, Forms Authentication, and Live ID
- Support for any SAML compliant provider e.g. Novell, CA Site Minder, Sun/IBM
- Federation allowing partners to access CRM using their existing identities

To put this in the context of the real estate application it would allow all the parties in a transaction to have access using their existing identities. A complete discussion of WIF and claims based security is beyond the scope of this paper but more information can be found at the MSDN[®] developer center at <http://msdn.microsoft.com/en-us/security/aa570351.aspx>.

Hosting Content in Windows Azure

In addition to being able to publish Dynamics CRM 2011 events to the Windows Azure Service Bus it is also possible to host content in Windows Azure. ISVs can build standard ASP.NET applications that deploy to Windows Azure and then are integrated into Dynamics CRM 2011 using an IFrame. Previously this used to be challenging due to the multiple security boundaries that are crossed in this type of configuration. However the new claims based authentication makes it possible to have a single sign on. This allows you to surface Windows Azure hosted content inside an IFrame of a Dynamics CRM 2011 form that is running either on-premise, partner hosted and even CRM Online.



Wrapping Up

The goal of this paper was to give an overview of the new capabilities of Dynamics CRM 2011 for building ISV applications. Dynamics CRM 2011 is a major feature release that continues to enable a broader set of applications to be built. In addition to the new features, Dynamics CRM 2011 has adopted the latest versions of the .NET Framework and Windows Workflow Foundation and other related Microsoft technologies allowing developers to leverage their core platform knowledge. ISVs building applications on Dynamics CRM 2011 are shielded from having to upgrade their own platform infrastructure and benefit from the commitment to backward compatibility. ISVs choosing between building on Dynamics CRM 2011 and directly on the lower level components should focus on the value of the "ripple" effect when deciding. If having services like security, workflow, reporting & more are automatically provided then the decision is simplified. Regardless of the type of application Dynamics CRM 2011 continues to expand the type of Extended CRM applications that can be built. Together, the core services of the platform form a powerful xRM Application Framework that ISVs can use to build powerful business applications.

Next Steps

Get Started with Microsoft Dynamics CRM today by visiting <http://crm.dynamics.com>. CRM 2011 is also available to ISV partners for early adoption and evaluation through the "Metro" program organized by Microsoft. Metro is a program focused towards early adoption of Microsoft technologies. Partners and customers in Metro can receive pre-release software, deep-dive training courses and showcase opportunities so that they can build and market their solutions on top of the Microsoft Platform. The Metro program covers early adoption of a number of Microsoft technologies such as .NET, Visual Studio, SharePoint, Dynamics CRM, etc. To join the Metro program, please contact your Microsoft representative or visit <http://connect.microsoft.com/metro>.

Microsoft Partners can also join the Microsoft Platform Ready program (<http://www.microsoftplatformready.com>) to get access to various resources such as training, support, testing and marketing resources to help you take your solution to market faster.

Microsoft Dynamics is a line of integrated, adaptable business management solutions that enables you and your people to make business decisions with greater confidence. Microsoft Dynamics works like and with familiar Microsoft software, automating and streamlining financial, customer relationship and supply chain processes in a way that helps you drive business success.

U.S. and Canada Toll Free 1-888-477-7989

Worldwide +1-701-281-6500

www.microsoft.com/dynamics

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, this document should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This White Paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2010 Microsoft Corporation. All rights reserved.

Microsoft, the Microsoft Dynamics Logo, Microsoft Dynamics, Outlook, Visual Studio, SQL Server, SharePoint, Windows Azure, Windows, and Windows Server are either registered trademarks or trademarks of the Microsoft group of companies.