

Microsoft Dynamics CRM 4.0

## Developing ISV Applications using Microsoft Dynamics CRM

**DAVID YACK**

May 2008

**Summary:**

Microsoft® Dynamics™ CRM 4.0 is built on a sophisticated line-of-business application platform that provides the basic services required by ISVs and developers to build upon. Clearly applicable in CRM-type scenarios, Microsoft Dynamics CRM also supports a wide range of business applications with the essentials required for building, delivering and maintaining them in a multi-tenant Software+Services environment. In this white paper we explore how, with Microsoft Dynamics CRM, an ISV can quickly build, deploy and manage line-of-business applications.

# Contents

<b>1. Introduction.....</b>	<b>4</b>
<b>2. Architecture Essentials .....</b>	<b>5</b>
2.1. Modeling Business Data.....	5
2.2. User Interaction .....	8
2.3. Integrating Applications.....	11
2.4. Business Process Automation .....	15
2.5. Platform Security.....	18
2.6. Extending with Web Services .....	19
<b>3. Deployment Essentials .....</b>	<b>21</b>
3.1. Software +Services – The power of choice .....	21
3.2. Packaging Solutions for Deployment.....	24
3.3. Wrapping up Deployment Essentials .....	26
<b>4. Operations Essentials .....</b>	<b>26</b>
4.1. Integrating Business Intelligence .....	26
4.2. Managing Operations .....	30
<b>5. What have we covered? .....</b>	<b>31</b>
<b>6. Where to from here? .....</b>	<b>32</b>

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, it 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.

Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, email address, logo, person, place or event is intended or should be inferred.

© 2008 Microsoft Corporation. All rights reserved.

Microsoft, Active Directory, , Excel, Internet Explorer, JScript, Microsoft Dynamics, Outlook, Powerpoint, SharePoint, Silverlight, SQL Server, Virtual Earth, Visual Basic, Visual C#, Visual Studio, Windows, Windows Live are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

## 1. Introduction

Microsoft® Dynamics™ CRM is a powerful business application platform that delivers to Independent Software Vendors (ISVs) an opportunity to extend and provide value added solutions. These solutions build on top of the robust core platform capabilities that Microsoft Dynamics CRM ships with, out of the box. Using these capabilities ISVs can quickly build both horizontal and industry specific vertical solutions without having to custom build the basic foundation.

Microsoft Dynamics CRM 4.0 continues the evolution of platform features that were available to ISVs in Microsoft Dynamics CRM 3.0. Many of the new features in Microsoft Dynamics CRM 4.0 will open the door for new types of applications and extensions to be built. Platform backward compatibility helps ensure that investments made by ISVs in the CRM 3.0 platform migrate forward to the 4.0 upgrade.

In this paper we will focus on key features of the platform that will allow ISVs to quickly build, deploy and manage an application built using the Microsoft Dynamics CRM 4.0 platform.

We have divided this paper into the following key logical areas:

- **Architecture Essentials** – The focus of this section is to help ISVs understand the core capabilities of the platform that make it easy for you to quickly build your solutions. In this section we will discuss the declarative data definition capabilities that are built-in to allow you to easily implement a reusable data model for simple and complex business data and the associated data relationships. Using this data model, the presentation capabilities make it easy to build the user experience including many options for extensibility.

We will then continue on to discuss the process automation model that offers numerous opportunities for ISVs to plug-in to the platform and customize processing. The platform also enables the custom definition of workflows using Microsoft Windows® Workflow Foundation. Platform security will be covered. And finally we will look at the robust web service SOA interface for further extensibility of clients that are both online and offline.

- **Deployment Essentials** – New to Microsoft Dynamics CRM 4.0 are expanded options for deployment to give the ISVs the *Power of Choice*. With Dynamics CRM, the ISV can choose to deploy on-premise inside the enterprise, or with a partner using the partner-hosted model (perfect for hosting vertical solutions where the partner can provide additional value added service), or simply using the Microsoft Dynamics CRM Online, a Microsoft-hosted SaaS (Software as a Service) subscription offering. This power of choice expands ISVs opportunities to reach customers interested in on-demand services by Microsoft. On the client side, end users also have the choice of using the Web Client, Outlook Client or the Mobile Client and can work both on-line and offline as needed.
- **Operations Essentials** – In today's fast-paced business environment reporting and analysis of business data is as important, *if not more* important than the collection of the data itself. Additionally, metrics and monitoring of the real time status of the platform using advanced system management integration is paramount to keeping today's 24/7 businesses moving forward. In this section we will explore the analysis and reporting capabilities of Microsoft Dynamics CRM. Using the combination of the platform and the advanced Business Intelligence capabilities of Microsoft SQL Server®, ISVs can create advanced data analysis capabilities to include in their solution offering. Additionally, we will explore new System Management capabilities that Microsoft Dynamics CRM 4.0 supports; allowing for enhanced

monitoring, including the monitoring of integrated ISV solutions to maximize solution availability.

The intent of this white paper is to give ISVs a high level understanding of the features and capabilities of the Microsoft Dynamics CRM 4.0 platform. Using this white paper, you will gain a basic understanding of the capabilities and use it to help you identify opportunities to build solutions on the platform. As you move past understanding the basics of the capabilities described in this document, additional resources such as the platform installation guide and the platform developer SDK will provide you the detailed knowledge necessary for you to fully implement your solution.

## ***2. Architecture Essentials***

In this section we will explore the core capabilities of Microsoft Dynamics CRM 4.0 that ISVs leverage as an application development platform to build solutions. One of the basic goals of the platform is to provide services and extensibility points that make it easier for ISVs to focus on their solution as opposed to building basic platform services (“The Plumbing”) like data modeling, security and other common services. The capabilities which we will discuss in this section, all serve as the foundation for building ISV applications and by using them ISVs will get from idea to implementation faster.

Along with having the core set of common capabilities that serve as the foundation for building solutions, ISVs also expect and demand flexibility to customize and/or extend the core capabilities by using plug-ins, workflows or other methods of altering how the core platform services work. This is an important part of the philosophy to enable ISVs to integrate custom processing. In this section we will also explore some of the business process extensibility points that are part of Microsoft Dynamics CRM 4.0.

### **2.1. Modeling Business Data**

Microsoft Dynamics CRM 4.0 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 SQL 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 CRM implementation. A number of standard data entities that are commonly known to exist in business applications built on the platform are pre-defined out of the box. Common entities like the Contact or the 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.

#### *Customizing Existing Entities*

Many of these system entities are customizable, 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 Company, School, or Hospital 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 customized for them, and not just one-size fits all.

### Defining Custom Entities

To provide flexibility, in addition to the customization of the system entities, the platform allows the ISV to define completely custom entities. This allows creation of entities to support the needs of any solution either vertical or horizontal in nature. A simple example of a custom entity might be in a real estate solution where an ISV could define a *RealEstateProperty* custom entity to track data about a specific real estate property in the solution. These custom entities are first class entities available for use with the entire extensibility capabilities of the platform. We will explore more of that as we go through the extensibility options later in this white paper.

In addition to using custom entities to store business data, custom entities can also be defined and used by ISVs to store information about the configuration or operation of their add-in or extension. For example, an ISV could define a *DataAuditOption* entity that could be used by a plug-in that performs data auditing. The custom entity would have attributes that would indicate how the plug-in would function. By using a custom entity for storing this, a separate database is not required and the ISV is able to leverage the platform to provide the user interface and other features such as offline synchronization. This also helps ensure that those configuration options are available even if the plug-in is running in offline mode.

### Easy To Modify

Changes to existing entities and the creation or modification of custom entities can be accomplished using either the Web Client Interface or the Metadata Web Service. Using the Web Client Interface a developer is able to define and modify entities and does not need to be a database professional with extensive database skills. The platform 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.

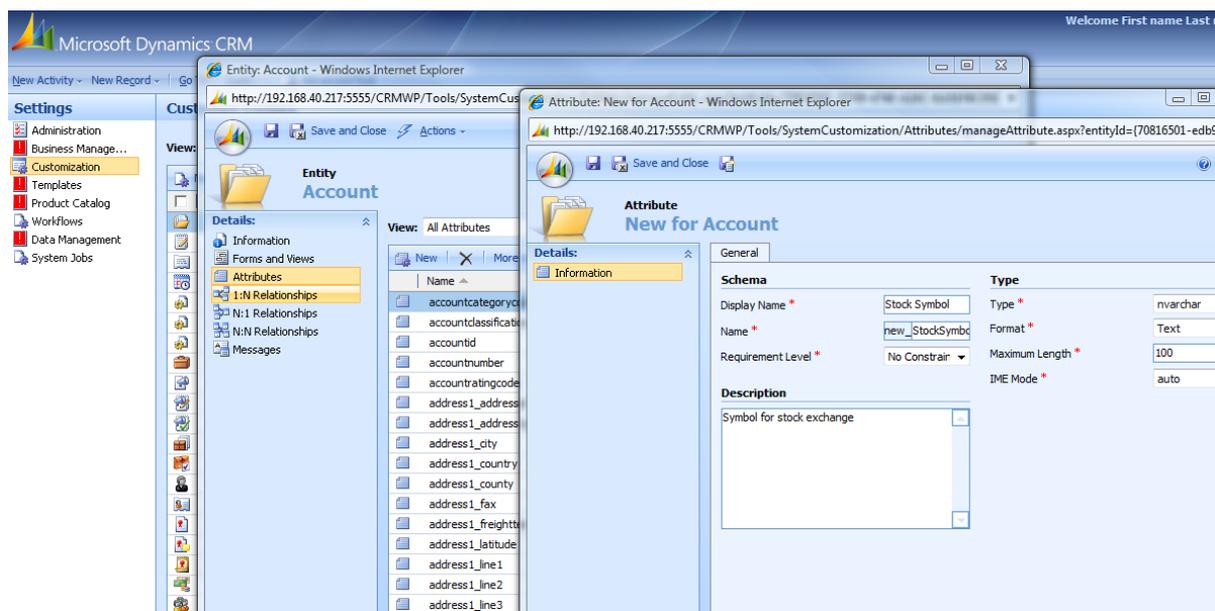


Figure 1 – The Microsoft Dynamics CRM Web interface for customization.

### Relating Data with Relationships

In order to allow modeling of real world applications, Microsoft Dynamics CRM 4.0 provides the concept of relationships. Relationships are used to define how one entity relates to another as well as to define the characteristics and behaviours of those relationships. Continuing the real estate example, using relationships you could establish a relationship that identifies which Contact is the property owner of the custom entity *RealEstateProperty* used in our example above.

The following are the type of relationships supported:

Relationship Type	Description
One To Many	Using our real estate example, a person might own one or more <i>RealEstateProperty</i> entities – we could define a relationship to track that business arrangement.
Many To One	Provides the ability to relate many items to one, a good example of this is the fact that each instance of a custom entity was created by a user, but that user could create multiple instances of the same custom entity. Each entity instance would have an N: 1 relationship with that user as its creator.
Many To Many	Provides the ability to relate one item to multiple items, and that one item to be linked to multiple items of the same entity type.  For example, we could define a custom entity called <i>PropertyType</i> it would contain all the possible property types our real estate <i>RealEstateProperty</i> property could be– we could then add a relationship to our entity in the system to be able to relate it to all the property types that match. More specifically, we might want to indicate it's both Office and Warehouse.  Using the many-to-many we could look at this relationship both ways – meaning we could ask the question what type of property is this...or which <i>RealEstateProperty's</i> are Offices .

The platform 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 platform 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 type 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. The platform also provides more advanced capabilities to control relationships that are referential and define how operations will cascade across their relationships. Using this advanced capability developers specify how they want the relationship is to be treated for various platform operations such as Share, Assign, Delete and Merge.

### Importing and Exporting Customizations

Modifications to entities, both system and custom can be exported for later reuse or deployment. With this capability ISVs specify domain-specific entities that are reused or otherwise packaged as part of their solution. We will discuss importing and exporting of customizations in greater detail in the deployment portion of this white paper.

### Data on the Go

One of the core capabilities of Microsoft Dynamics CRM 4.0 is the ability for data to be synchronized 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 synchronized by a user using the Microsoft Outlook® CRM Add-in. Microsoft Dynamics CRM 4.0 manages the synchronization of data; including any changes the user makes while offline from the host CRM server. The developer can identify which entities are available offline and the user can control if, or how much of a specific entity they take offline.

### *Wrapping up Data Modeling*

By having a flexible data modeling and data management capability as part of the core platform, ISVs quickly define domain data and the relationships between that data. ISVs also benefit from allowing the platform to manage the data, instead of requiring a separate SQL database. In the next section we will see that the interaction with the user on that data is also easily customizable. Additionally, we will see that this data model will be further used for extensibility when we talk about workflow, plug-ins and other extensibility options.

By allowing the platform to manage the data model, the platform is able to provide a rich set of features like de-duplication, offline synchronization and a full web service interface even to custom entities. It would take extensive development work for an ISV to build each of these features from scratch.

As we continue forward and explore more of Microsoft Dynamics CRM 4.0 we will see how the data modeling and management capabilities are part of a rich platform that can serve as an application development platform for building any number of ISV solutions.

## **2.2. User Interaction**

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 4.0 provides the user experience using HTML-based pages to browse and manage data. ISVs have the ability to customize many aspects of the user experience including form content and site navigation. Interaction with the user can also be customized by events exposed by the platform UI Engine and custom ISV-provided scripting. External applications can also be integrated as part of the navigation as well as embedded inside CRM forms.

The user interacts with Microsoft Dynamics CRM 4.0 pages using the Web Client Interface or from within Microsoft Outlook. The Web Client Interface uses a standard web browser and requires the user be on-line with access to the host server. On the other hand, the Microsoft Outlook user experience has the capability being used offline in addition to the same online capabilities provided by the Web Client. Regardless of whether the user uses the Web Client Interface or Microsoft Outlook the pages are displayed to the user in the same format (HTML) providing a more consistent experience for the user.

Each entity defined will have an associated main form that is used for viewing detail on the entity and for data input. The following is an example of an entity form for the built-in Account entity.

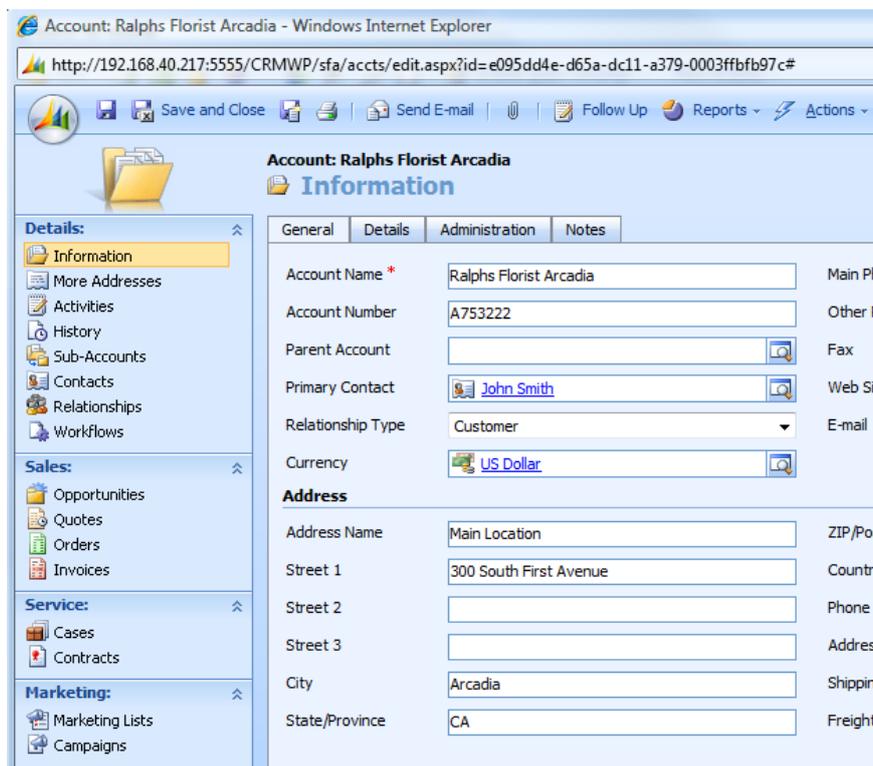


Figure 2 – The standard Main form for the Account Entity.

Additionally, each entity will have one or more views that provide a grid view of the data. The views are customizable in both what columns show as well as the criteria for what subset of the entity's data will qualify for display in the grid. Views can be defined at the application level. Additionally, the platform provides users the ability to create their own custom views from within the user interface.

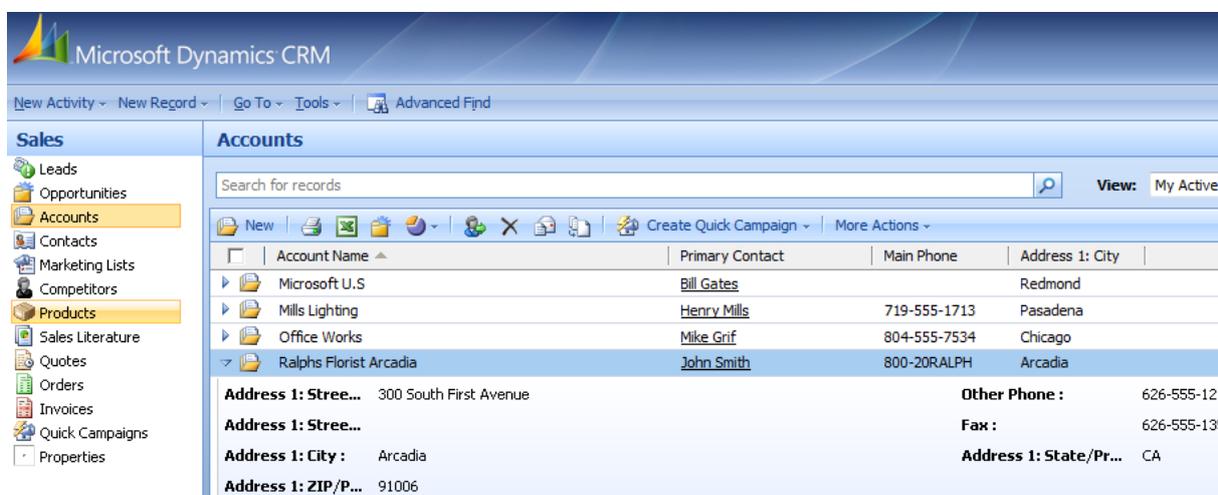


Figure 3 - The results grid and the customizable views for the Account entity

### Customizing Forms

Customizing forms is quite simple as the Dynamics CRM platform has provided a number of useful and flexible objects. The entity forms contain a set of Tab Panels. The developer can also add additional tabs. Each of the Tab Panels contains one or more Form Sections. The developer may control and configure the layout of the sections within the Tab Panel. Each Form 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 customized using the form editor.

The following image shows the web based form editor used by the developer to modify the forms.

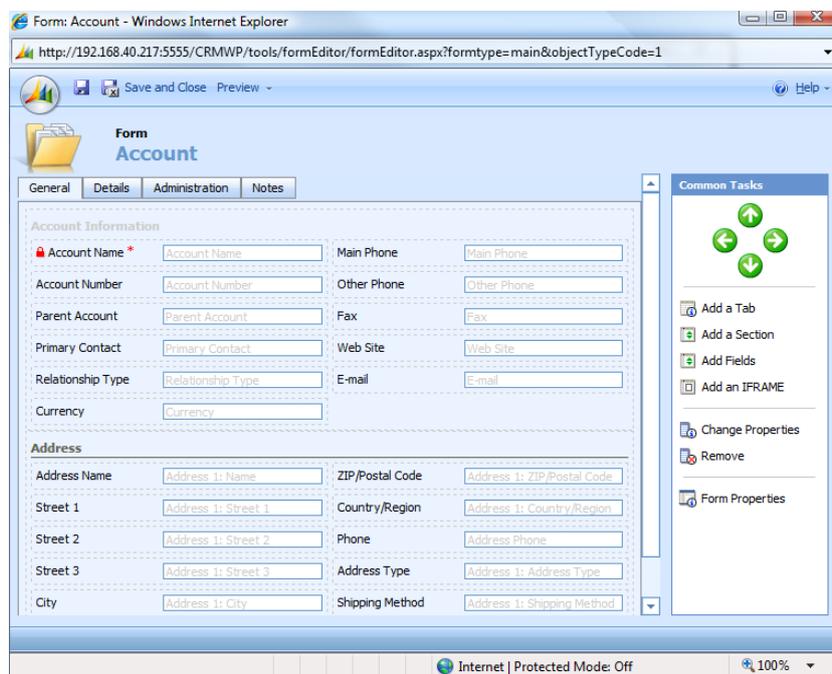


Figure 4 – The Microsoft Dynamics CRM Form Editor is used to customize forms

### Publishing and Import/Export

Any changes made by the developer to the forms are not visible to the user until explicitly “published”. This helps ensure that the user does not see partial changes and allows the developer more flexibility in controlling when changes appear. Similar to the data model customizations, changes to forms and views are also exportable so they can be re-imported and reused many times as part of an ISV deployable solution.

### Client Side Programming

As part of the entity form the platform provides the ability for ISVs to run custom scripts when events occur both at the entity form level and field level inside the form. At the form level, events like OnLoad and OnSave are provided for customization. Each field on the form also exposes an OnChange event that can be used to react to modification to an individual field.

The custom scripts that are triggered by the events can perform tasks like setting default values, performing client side validation or calling out to 3<sup>rd</sup> party services. Since the forms are rendered in HTML, JScript<sup>®</sup> is used as the scripting language.

To help promote upgradeability and supportability of the client side customizations, Microsoft Dynamics CRM 4.0 provides a client side object model and a related set of helper methods for the

script to interact with the form and its contents. For example, rather than forcing the developer to find a field on the form by navigating the HTML, a global form variable CRMForm is provided that the script can use. To access the value of the Address1 attribute, the following would be used:

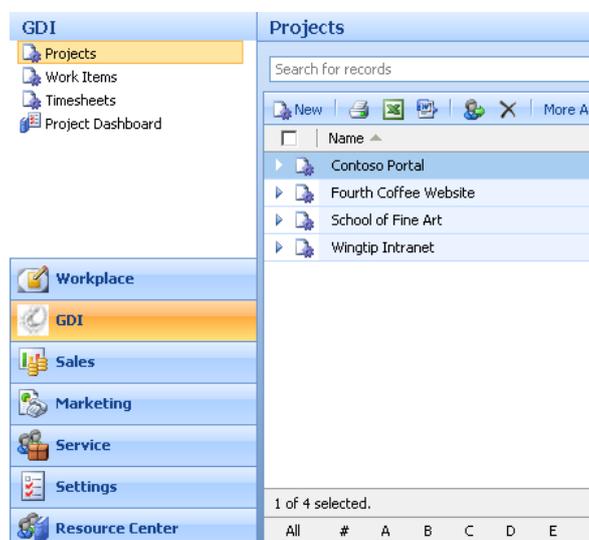
```
CRMForm.all.new_address1.DataValue
```

Additional helper methods and properties are provided to allow detection of whether the form or even specific fields are modified and other useful information that can be used in writing the script.

Using client side programming, ISVs can provide more real-time interaction with the user without having to build custom forms.

### *Site Navigation Customization*

It is unlikely that any two applications built on the Microsoft Dynamics CRM 4.0 platform will have the same demands for site navigation. The navigation model supports this need by allowing ISVs the ability to integrate other web applications and to modify the default navigation. The SiteMap provides the definition of the navigation structure. By modifying the SiteMap an ISV can control the entire navigation structure and the ordering of subareas within the site navigation. This includes the ability to place custom entities into the navigation hierarchy in specific locations.



*Figure 5 - Site navigation customized by the ISV to show custom entities*

## 2.3. Integrating Applications

Combining existing applications or adding custom application functionality into the platform provides users with an improved user experience by avoiding the need to open several applications separately. This can be accomplished in a few ways, the simplest being modification of the site navigation to include links that launch external applications. It's also possible to have deeper integration by using form customization to embed external applications into the context of an entity Form and have it appear as if it was part of the application the user is interacting with. Context information can be passed to the custom application to allow it to be context sensitive to the data being used by the user in the rest of the form.

### *Simple Links*

Adding a link is an easy way to launch an external application. A more powerful capability is providing a link in the context of an entity form. For example, by providing a link to the shipping company tracking page on the Order form the user could easily lookup the tracking information. This would be much easier for the user than having to copy the tracking number, launching the tracking application and paste in the tracking number and request the status.

Another example of this type would be launching a SharePoint® workspace related to a platform entity. In this scenario assume that you want to leverage the document workspace capabilities of SharePoint to share documents with check-in/check-out capabilities. On the Opportunity Entity you could add a field to store a link to the SharePoint document workspace. Upon creation of an Opportunity entity a plug-in could be used to call the SharePoint API to create the workspace and store the URL in the field. Then when the user views the opportunity they could simply click on the link to launch and load the document workspace. The goal is to remove user friction and leverage as much pre-built capabilities as possible, making it easier for the user to accomplish tasks in fewer steps.

### *Simple ASP.NET Extension*

ASP.NET can be used by ISVs to build extensions that plug right into the platform using IFrame extensibility (or via a direct link to the page). This gives the ISV the complete power of ASP.NET and the ability to build their applications using Visual C#®, Visual Basic® .NET or their choice of supported managed code. Using IFrame integration, the ASP.NET application can be passed context information such as the ID of the platform entity that is being shown to the user on the form. The ASP.NET application can then do any type of processing necessary and interact with the user. This interaction can include having the ASP.NET application call the platform web services to obtain or modify additional data.

While the built-in forms that the platform offers provide a quick and flexible way to have the user modify platform data; using ASP.NET web forms allows other non platform data to be integrated on the same area seamlessly. Additionally, new capabilities added by ASP.NET AJAX allow ISVs to take advantage of features such as partial update and improved client interaction. These capabilities can make things like dynamic loading of drop downs or calling back to web services from the client side easier to perform.

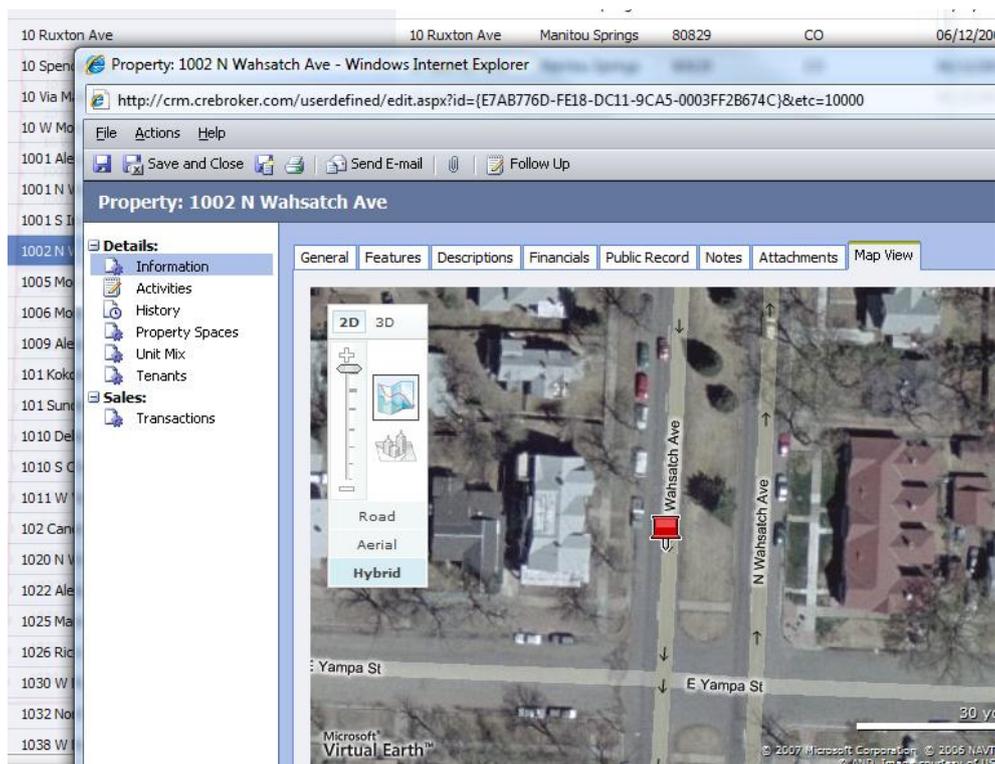
Also new to Microsoft Dynamics CRM 4.0 is the ability to access the platform web services when the application is offline. The platform enables this support by providing a more consistent web service interface for applications running on the client machine to use when it's offline. This web service is hosted on the client using a light-weight web server as its hosting platform. This support is important because usually when using the ASP.NET capability with IFRAME, by default it requires that the client be online or able to access the server hosting the ASP.NET application. Using the new offline support ISVs now have the ability to place a copy of their ASP.NET application on the local client machine, hosted by the same light-weight web server and access the platform data using the offline web service interface. This provides the user similar access to the ASP.NET extension regardless of the user being online or offline.

### *Interoperating with other SaaS platforms*

Another great way to extend the capabilities of the platform is to use it with other SaaS offerings. A common example is to add maps using Virtual Earth™. Virtual Earth is hosted on Microsoft's servers and offers interaction using either an API or via the map control that users can interact with directly. Virtual Earth can be used with the platform in ways that are not visible to the user or used as a data visualization tool to show geographic location information on data stored in the platform.

An example of using Virtual Earth behind the scenes is validation or geo coding an address. Using the form event capabilities, you may add JScript code that is called during the OnSave event of the form. In this code you could call Virtual Earth passing the address – if Virtual Earth wasn't able to recognize it and return a match you could prompt the user to confirm the address. In this example we are talking about Virtual Earth, but this same technique could be used to integrate with any service including one built by an ISV.

For a more visual example, you could add a tab to a form that would contain a Virtual Earth map. A great example of this is in a real estate implementation where you have a Property entity and a tab is added that contains the map. The map then could have a push pin indicating the location of the property. More advanced data could also be shown on the map to indicate the income and population of the area the property is located. From a user perspective, they would perceive the map as being a seamless part of the application. The following is an example of the Property form with the map.



**Figure 6 - Seamless integration of other SaaS applications such as Microsoft Virtual Earth is easily accomplished with Dynamics CRM**

### Leveraging Silverlight

Microsoft Silverlight™ is a cross-browser, cross-platform implementation of the .NET Framework allowing development of rich interactive applications for the web. Using Silverlight in conjunction with Microsoft Dynamics CRM 4.0 opens up opportunities to offer customizations that include rich media capabilities. Silverlight can be used with the platform by using the IFrame customization capability. Using an IFrame on a form you can include rich islands of information or applications. Silverlight specializes in rich content, such as video, animations, and 2-D graphics. A simple example of how you might use Silverlight would be to use it to visualize some data like purchase probability by showing it with some comparison pie charts. Using Silverlight you could host this in an IFrame associated with the customer or opportunity. Additionally, using the media capabilities you could include videos of the prospects requirements gathering as a tab on the opportunity in order to make

it easy to reference. Silverlight media playing capabilities could provide custom controls embedded directly into the standard form.

Silverlight is built to be hosted in a browser html document, and allows the hosting document to interact with it using JScript. The actual contents of the Silverlight area are defined using Extensible Application Markup Language (XAML). End users' browsers require a small plug-in (1-3mb) that must be installed on their local machine. This is a fast and simple process.

As an example of how Silverlight can work with the platform capabilities let's explore the need to integrate some rich content with our product list. Imagine that you need a simple way to allow your sales staff to get up-to-speed on the specifics of each product. You could add a tab to the standard product form that provided video content hosted in an IFrame with a Silverlight application. Leveraging Silverlight's media capabilities makes it easy to play videos and other media content. Silverlight Streaming is a hosted offering by Microsoft that can be used to host Silverlight media applications and their content for easy playback and use.

To illustrate how simple it is to use the media capabilities – the following is a quick example of the minimal XAML required to use the MediaElement:

```
<Canvas
  xmlns="http://schemas.microsoft.com/client/2007"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  Width="300" Height="300">

  <MediaElement x:Name="myMedia" Source="CustomereVideo1234.wmv"
    Width="500" Height="400" />

</Canvas>
```

While that would provide the simplest path to playing back a video, it doesn't leverage the capabilities of the MediaElement by providing custom controls or leveraging the Timeline Marker capabilities that could be used to detect reaching a certain part of the video and showing helpful tips or other content to the user.

Another example of how Silverlight can help extend Microsoft Dynamics CRM 4.0 is by looking at Silverlight's ability to support ink input from computers with pen digitizers (e.g. Tablet PCs). Standard forms in the platform really aren't designed with any special ink input support, however, by combining a Silverlight panel, it is possible to provide a tab with a custom control specifically designed to make input easier for pen or touch input users.

Using the ink capabilities of Silverlight integrated within the standard form you can use it to add handwriting, drawing, annotation and navigation into your standard forms. Silverlight provides a complete set of objects including the InkPresenter that facilitate the ink support.

All of the capabilities of Silverlight described so far exist as part of the 1.0 release. Silverlight 2.0 (available today in beta form) represents the next evolution in the technology but more importantly a strategic move as it opens up the ability to run managed code in the browser for the first time. This allows developers to use languages like C# instead of JScript and will result in a substantial increase in performance of these rich applications.

In this discussion of Silverlight we have only touched the surface of its capabilities. The key take away is that Silverlight opens the doors to a number of new opportunities for integrating richer content and new types of user interaction that haven't been available with the platform in the past. As an ISV this represents an opportunity to substantially enhance your solution leveraging the pre-built capabilities of Silverlight to do it much quicker.

### Microsoft Office Interoperability

Out of the box Microsoft Dynamics CRM 4.0 works well with Microsoft Outlook and a user can use that as their primary interface to the platform. All the form customizations discussed earlier work in the Outlook client both online and offline. Visual Studio® Tools for Office (VSTO) is another opportunity for deeper integration that ISVs could use is to build additional functionality. Using VSTO, ISVs could build applications that extend and leverage the capabilities of Microsoft Word, Excel®, PowerPoint® or Outlook and access the platform for the data content using the web service API.

## 2.4. Business Process Automation

One of the challenges of building ISV solutions or line of business applications on top of an off the shelf platform is the difficulty of integrating custom business processes into the heart of existing platform processes. Microsoft Dynamics CRM 4.0 addresses this problem head on with support for Plug-ins and a robust workflow capability. These two capabilities are implemented using a unified event framework. The following diagram shows how plug-ins and workflows fit into the platform architecture.

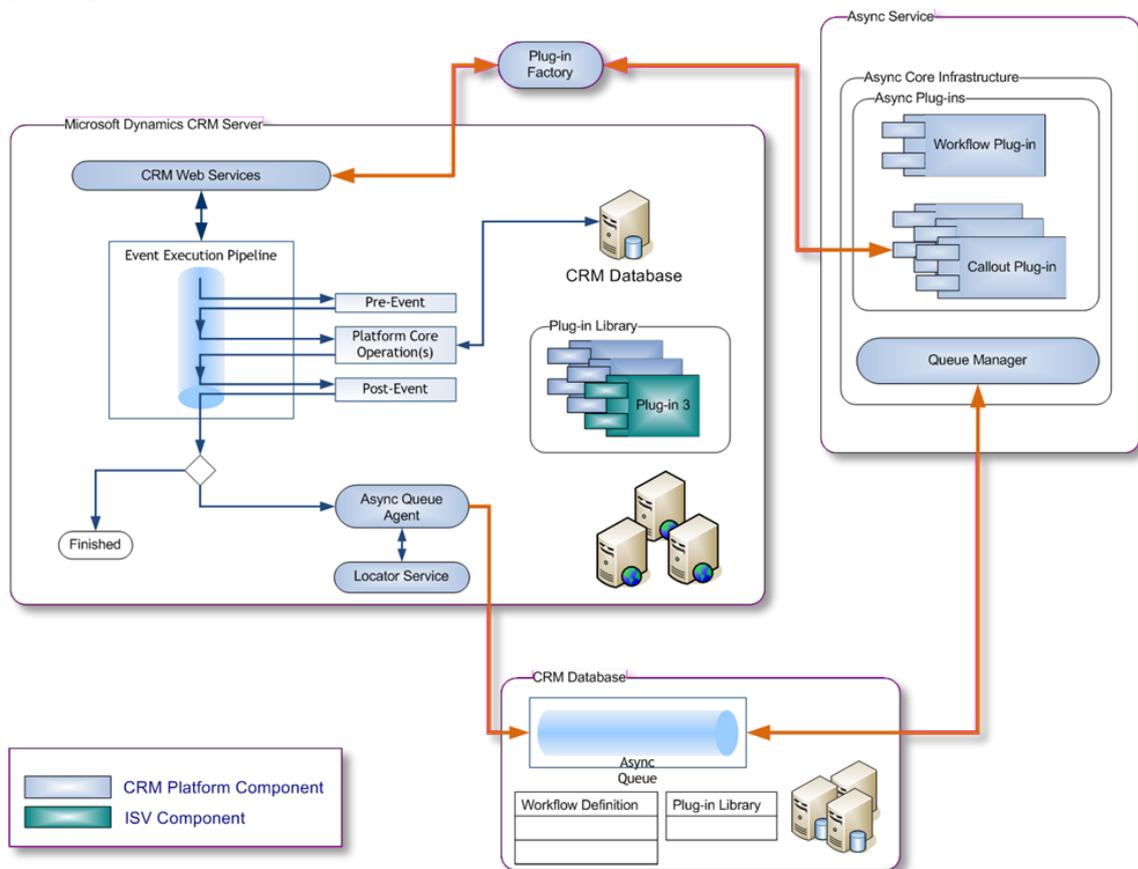


Figure 7 - Dynamics CRM provides a unified event execution pipeline allowing comprehensive extensibility with Plug-Ins or Windows Workflow Foundation.

### *Plugging in with Plug-ins*

Plug-ins provides the opportunity for ISVs to insert themselves into platform services and grab control before and/or after a platform operation is performed. Take the example of a user adding an account record. When that user presses the save button, the UI will contact the platform and make a Create request. Using Plug-ins it is possible to register ISV 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 platform create was processed. ISVs can also register to be notified after a platform operation is completed. This type of use might be appropriate for auditing, for instance. A simple way to think of this is that the platform goes through an Execution Pipeline when processing a request. Using plug-ins you are able to insert your custom logic into the Execution Pipeline.

Plug-ins are considered an extension of the platform and trusted code. They are written in a .NET language such as VB .NET or C# and implement a simple plug-in interface (IPlugin) that requires implementation of a single method. Using the plug-in registration tool or the plug-in registration API, plug-ins are associated with platform events (e.g. Create, Update etc.).

Plug-ins can operate as synchronous plug-ins meaning that their action happens before the Execution Pipeline completes or Asynchronous using a fire and forget model. Synchronous plug-ins are best used when the custom processing needs to be completed before control is returned to the user either via the user interface or web service interface. For this reason, synchronous plug-ins are best used for short duration processing that does not involve lengthy operations. Asynchronous plug-ins on the other hand are more geared toward getting something done, but not making the user wait. Auditing is a good example; it needs to be done, but not necessarily prior to giving the user control back.

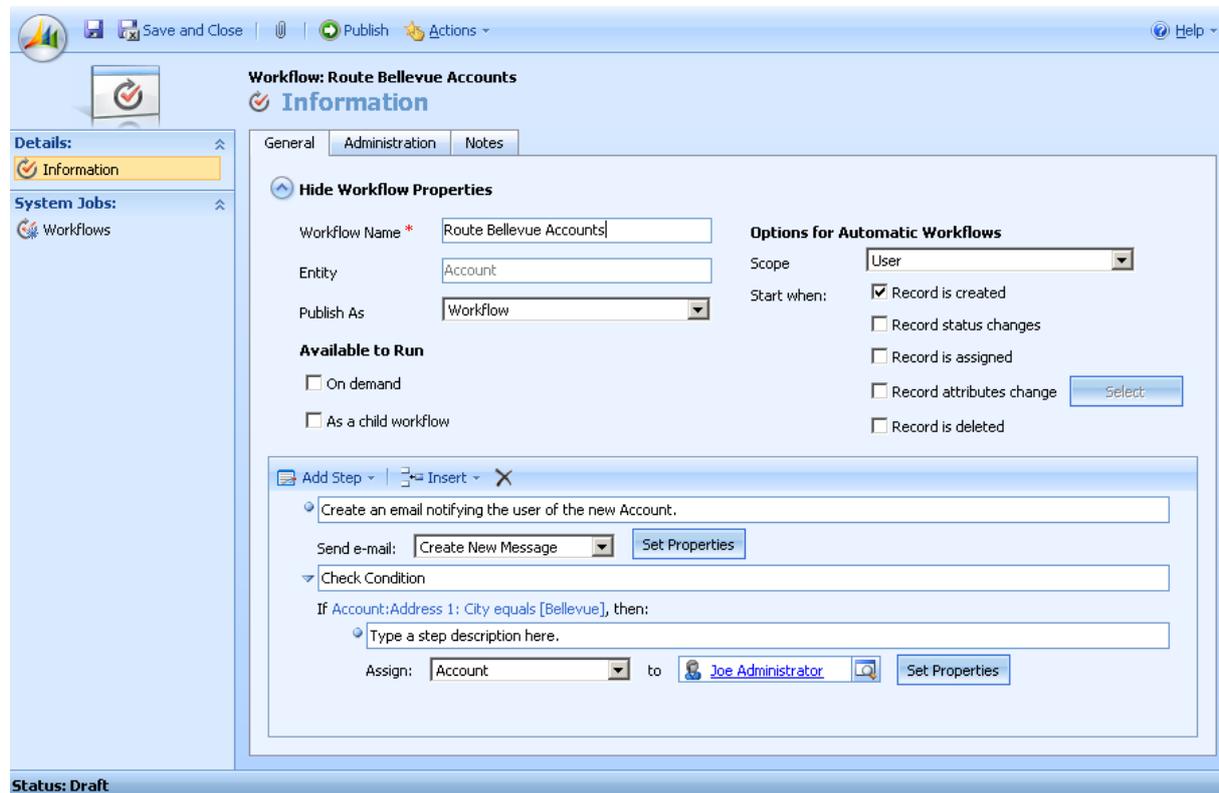
Plug-ins have the ability to execute both on the server when requests are made or they can be provisioned to run offline. Offline support is used to facilitate having custom logic run when a user is using the Microsoft Outlook CRM Client and is offline. Imagine building an offline application (to be used by sales people on the road, for example) and you need to implement custom validation that can't be performed using the form customizations. When the user is back online and able to synchronize, 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 sales person would get that error immediately even when they are offline because that plug-in would fire in the offline client.

### *Automating with Workflow*

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

Workflows are similar to plug-ins in that they participate in the same unified event model and are processed and managed by the platform services. As a developer you don't have to worry about the mechanics of how workflow is implemented. You focus only on building the workflows that run on the platform. Workflows are different from plug-ins because they are more visible to the user. Plug-ins execute 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 in 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.

Using the Web client, 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. The following screen allows the user to define their own workflows.



*Figure 8 – Dynamics CRM provides a web client that may be used by a business analyst or end-user to define or modify workflows as business needs change.*

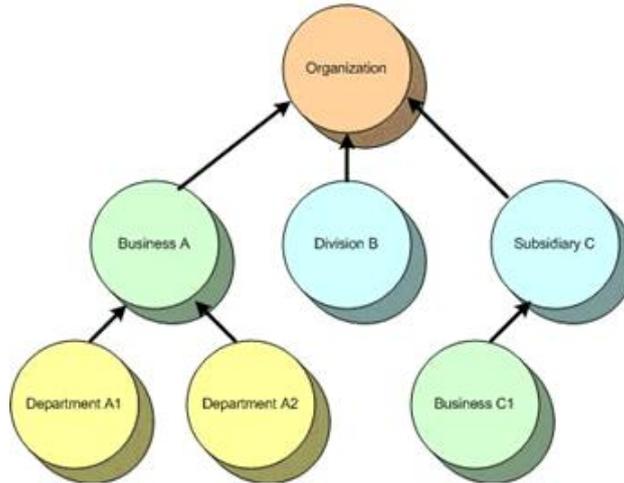
In addition to the list of steps available out of the box, ISVs can create “Custom Workflow Activities” which can be added to the lists of available steps. Custom activities are .net assemblies that can be created using Visual Studio. These activities may either inherit from the built-in capabilities adding custom logic or be completely customized. Building custom activity libraries is a great way to package up ISV capabilities. These libraries of activities may be used internally by the ISV to make building the ISV solution easier. Additionally, a library of ISV solution activities could be provided to customers to allow them to further customize the ISV solution by building custom workflows using the ISVs pre-built workflow activity libraries. For example, an ISV could provide a Customer Purchase Probability Scoring workflow activity that predicted the likelihood of a customer making a purchase. That activity could expose properties that would be set so that it could evaluate the customer and produce the probability score. A customer of the ISV’s solution could then use that activity on a custom workflow or from within the Web Client interface to calculate the score activity as part of an opportunity workflow.

#### *Wrapping up Business Process Automation*

The Plug-ins and Workflow capabilities of the platform provide deep integration capabilities for ISVs building solutions. Both techniques offer unique capabilities and keep the ISV focused on the solution as opposed to the platform plumbing that would be required if the ISV had to implement both these services on their own.

## 2.5. Platform Security

Security plays an important part in any application platform 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 4.0 provides a rich security model that applies to the built-in application and platform components but also to the customizations you build as part of an ISV solution.



*Figure 9 – A typical organization hierarchy with a number of business units*

Access to the platform itself is controlled by a pluggable authentication model. Using this model the appropriate authentication provider is used based on the deployment model chosen at installation time. Currently, support is provided for Active Directory<sup>®</sup>, Forms Authentication and Windows Live<sup>®</sup> ID. Authorization identifies the user to the platform and therefore controls the access entitlements the user receives.

Platform authorization is responsible for controlling access to the data and services using a role based security model. A user is assigned one or more roles that aggregate platform security entitlements to provide a simple answer to the basic questions - what can this user do and what data can they access.

To provide additional flexibility the concept of business units is used to group users and grant authorization based on the roles. Each installation has the concept of an organization; the business units exist within the organization 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 / organization 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.

A role can specify the user's entitlements to create, update, delete, share and assign for each entity. An administrator can also further refine those entitlements to grant that organization wide, business unit wide or only to user's own data.

Customization of security roles is done using either the user interface, the import/export capabilities or via the platform APIs. This means that as ISVs build their solutions on the platform they can deploy with their solution a set of roles that are custom to that solution. During installation time, these can be provisioned on the target organization and then assigned by the system administrator as users are created.

Role: CEO-Business Manager									
Entity	Create	Read	Write	Delete	Append	Append To	Assign	Share	
Account	●	●	●	●	●	●	●	●	●
Contact	●	●	●	●	●	●	●	●	●
Lead	●	●	●	●	●	●	●	●	●
Opportunity	●	●	●	●	●	●	●	●	●
Activity	●	●	●	●	●	●	●	●	●
Note	●	●	●	●	●	●	●	●	●
E-mail Template	●	●	●	●	●	●	●	●	●
Announcement	●	●	●	●	●	●	●	●	●
Subject	●	●	●	●	●	●	●	●	●
Queue	●	●	●	●	●	●	●	●	●

Figure 10 - Microsoft Dynamics CRM provides a rich security model used by both built-in applications and ISV solutions

Once defined the platform security affects the user experience via the user interface (both on-line and offline) as well as when that user accesses the platform using the APIs and when using Filtered Views. This helps in providing a more consistent approach to security and frees you as an ISV to focus on the solution implementation. System administrators are provided a familiar interface to define roles and interact with the security provided by your solution.

## 2.6. Extending with Web Services

Microsoft Dynamics CRM 4.0 provides a dynamic web service interface for applications to use to access and manipulate platform data as well as interact with platform services. These services allow ISVs to write applications using Visual Studio or other developer tools by simply referencing the platform web service. The web services exposed by the platform are WSI BP 1.1 compliant. This compliance support makes the web services interoperable with non-Microsoft platforms.

As you customize the data model by adding custom entities, custom attributes or additional relationships the web service interface is dynamically adjusted to provide typed interfaces to your newly defined customizations. Each entity may then be accessed via web services using standard Create, Read, Update and Delete requests.

### Using the Web Services

The following is an example of using the web service interface to create a new account.

```
// Assuming you already have a configured CrmService class named "service"
// Create an account entity and assign data to some attributes.
```

```
account newAccount = new account();
newAccount.name = "Mountain Retail Store";
newAccount.address1_postalcode = "80920";
newAccount.address1_city = "Colorado Springs";
```

```
// Call the Create method to create an account.
Guid accountId = service.Create(newAccount);
```

This same method is provided not only for entities built-in but as the data model is customized the same capabilities are extended to the custom entities. This provides a more consistent interface that is based on the platform metadata and strongly typed. It is dynamically configured as you alter the platform metadata. Developers then have one technique to learn in order to manipulate any of the platform entities both built-in and custom.

#### *SOA Helps Ensure Consistency*

In using a service oriented architecture (SOA) the platform helps ensure consistent validation of requests for modification of the data. It also enables plug-ins and workflows to run appropriately when data is modified using any user interface as well as any API to the platform. This consistency helps maintain a high level of integrity and security of the data regardless of how it is being accessed or manipulated.

ISVs may use the web service interface when they build plug-ins, custom workflow activities as well as complete custom applications that interact with the platform. For example, an ISV could build a standalone application leveraging Windows Presentation Foundation (WPF) that provides a different look and feel for the user experience but under the covers uses the platform web services.

#### *A Business Application Platform – Not Just CRM*

In this section we explored the business application platform capabilities of Microsoft Dynamics CRM 4.0. While the name has CRM in it, and clearly it's applicable in CRM-type scenarios, the platform can also be leveraged to build a wide range of business applications. Additionally, time to market is critical for ISV solutions so we tried to highlight the capabilities of the platform and how it provides the basic plumbing so you can focus on the solution you're trying to build.

Each of the capabilities provided by Microsoft Dynamics CRM 4.0 fulfills a need that ISVs would have to otherwise spend hundreds of hours building on their own. For example, ISVs could choose to implement Windows Workflow Foundation on their own without a platform like Microsoft Dynamics CRM 4.0 and with 3 or 4 lines of code the ISV could start-up the Windows Workflow Runtime. A few more lines of code and the ISV could fire up a workflow instance. While that is certainly a possible option, it fails the real world implementation test of requiring a robust, reliable and manageable solution. Using Microsoft Dynamics CRM 4.0 the ISV gains all that out of the box – workflows are started and run on one or more servers and can even be clustered to provide scalability. When a workflow fails it can be detected not only by the system administrator but by any user that has that privilege. In fact users can not only restart a failed workflow, they can create and define their own using the platform Web Client.

While we could go on with a list of what an ISV would have to implement on their own, the reality is that to provide end to end integration is a much more complex task and by using a platform like Microsoft Dynamics CRM 4.0 those complex tasks are already provided by the platform out of the box.

### ***3. Deployment Essentials***

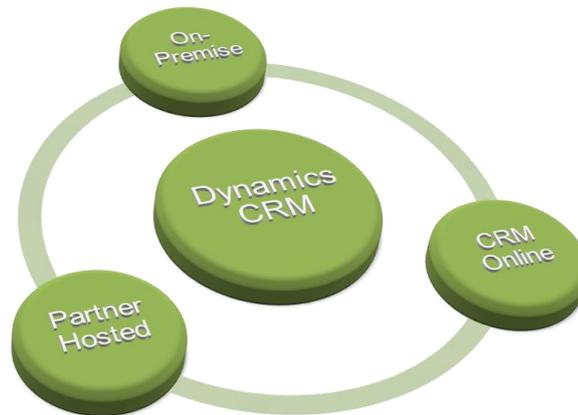
In this section of the paper we will explore platform deployment options. Microsoft Dynamics CRM 4.0 offers ISVs, partners and end users the power of choice in determining many aspects of a deployment.

The following list highlights some of the key choices and options:

- **Choice of Client Access** – The platform can be accessed by users using Microsoft Outlook, Internet Explorer®, the mobile client or a completely custom client application. This allows users to choose what works best for them at any point in time. Depending on user needs deployments can include a mix of these.
- **Choice of a Server Location** – Microsoft Dynamics CRM 4.0 can be deployed on-premise at a customer location, at a partner hosting facility or in a Microsoft datacenter using the Dynamic CRM Online service offering.
- **Choice of a Licensing Model** – Customers can choose a licensing model that is appropriate for their business and use of the platform. They can choose from standard licensing or a subscription where they pay monthly; essentially renting the software with no long term financial commitment. ISVs also have the option of embedding or bundling the platform with their product and paying royalties with each customer deployment.
- **Ability to change your mind** – In today's dynamic business environment, flexibility and choices are great, but ability to change your mind is even more important. With choices of client access, server location and a licensing model the platform leaves many options open. The same software runs each type of deployment giving the user the ability to move between them as needed to support their business.

#### **3.1. Software +Services – The power of choice**

For years the obvious choice for software deployment was on-premise in your own data center inside the corporate walls. Today however, new options are emerging as bandwidth and security are no longer the roadblocks to hosting solutions externally. In fact, for many companies the boundary of their organization's IT Infrastructure is much more virtual and includes services that are hosted by service providers. These loosely-coupled distributed systems may be integrated together to provide an organization's complete IT service capabilities. Microsoft Dynamics CRM 4.0 provides three distinct choices for where the server portion of the application may be deployed – On-premise, Partner Hosted or CRM Online.



#### *Understanding On-Premise Deployments*

In the on-premise deployment scenario the platform is deployed in the customer data center. This is the most traditional deployment model. All capabilities of the platform are available including managed code, custom workflow activities and plug-ins. User authentication is driven with Active Directory and users are typically associated with the Active Directory infrastructure.

#### *Understanding Partner Hosted Deployments*

In the partner-hosted deployment scenario the platform is hosted by a partner at their datacenter facility. In this deployment model partners can leverage the platform in conjunction with Microsoft Host Messaging Collaboration (HMC) or as standalone installations. Using HMC provides processes for automation of provisioning of organizations and new servers. All capabilities of the platform are available including managed code custom workflow activities and plug-ins as long as the partner enables these capabilities. User authentication is done via Forms Authentication from the user point of view, but behind the scenes it is talking to the partner active directory infrastructure.

#### *Understanding CRM Online Deployments*

In the CRM Online deployment scenario the customer accesses the solution at the Microsoft datacenter using CRM Online services. The CRM Online hosting environment makes it easy for customers to get started quickly without having to invest in and setup on-premise hardware because it is provided and managed by Microsoft. Most capabilities of the platform are available; so customers get a fast, flexible and familiar CRM solution. For reasons of security and integrity some types of customizations and development will be restricted on the CRM Online platform. Later we will have a more detail comparison table that will show you specifically what features for customization are available in each of the deployment scenarios. There are a number of ways ISVs can leverage the CRM Online platform as part of their broader CRM solution strategy.

The CRM Online scenario offers the opportunity for ISVs to build on the platform building blocks that include the data modeling and UI customization capabilities that are present in all solutions. The platform can also be leveraged as the connector and aggregator of other Software + Services/SaaS offerings that are complementary. Using this approach CRM Online becomes the user's portal to interact with a number of services that act together as one from the user's perspective. CRM Online, having most of the capabilities of the other deployment scenarios, offers a rich platform for building complete solutions.

Additionally, the fact that the data model customizations, custom client scripting and workflow to name a few will be aligned in all the deployment scenarios set the stage for ISVs that have an existing on-premise or partner hosted solution to build an entry level version of the solution that targets CRM Online. Customers could then use your customizations to get the feel for the platform and start adoption in their business. As their needs graduate to needing more process automation or other customizations that can only be done using plug-ins and custom code workflow activities, they could migrate to an on-premise or partner hosted environment. This type of migration could open the door for the customer to upgrade to the more sophisticated version of your solution as well.

Another example of an ISV opportunity is to leverage the Web Services exposed by CRM Online to completely integrate CRM capabilities into an existing ISV application. The Web Services in CRM Online are the same as the other deployment scenarios except they don't have provisioning type access that isn't appropriate for a hosted environment. In this scenario the ISV would leverage the platform capabilities by using it as a software service in the background of an existing client application. This would be a great way for an ISV to add traditional CRM type capabilities to their application. This could even be used for non CRM scenarios to simply leverage the easily configured capabilities of the platform and to provide data storage in a reliable cloud in the sky

ISVs can also build mashups that integrate external applications and services using IFrame and other scripting techniques so they look like they are part of the platform. The full client side scripting capabilities are available in CRM Online so these types of customizations migrate easily between the deployment scenarios.

Even if you are not building a CRM Online solution initially, take the time to understand the opportunity and the capabilities of the platform. That way as you build your solution you can consider leaving place holders for offering a CRM Online version.

### *Three Options – One platform*

Regardless of the deployment scenario it's the same platform in each model. The following table should help you understand the differences with respect to the development and customization tools available in each of the three platform options.

Type	Detail	On – Premise	Partner Hosted	Online CRM
Basic Customization	<ul style="list-style-type: none"> <li>• Entities</li> <li>• Attributes</li> <li>• Forms / Views</li> <li>• Relationship</li> </ul>	<b>YES</b>	<b>YES</b>	<b>YES</b>
Business & Data Modeling	<ul style="list-style-type: none"> <li>• Business Units</li> <li>• Teams</li> <li>• Security Roles</li> </ul>	<b>YES</b>	<b>YES</b>	<b>YES</b>
Controlling & Customizing UI	<ul style="list-style-type: none"> <li>• Navigation (sitemap)</li> <li>• Web client Jscript</li> </ul>	<b>YES</b>	<b>YES</b>	<b>YES</b>
Import & Export Data	<ul style="list-style-type: none"> <li>• Entities</li> <li>• Attributes</li> </ul>	<b>YES</b>	<b>YES</b>	<b>YES</b>

Type	Detail	On – Premise	Partner Hosted	Online CRM
Import & Export Configurations	<ul style="list-style-type: none"> <li>• Customizations</li> <li>• Workflow</li> <li>• Sitemap</li> </ul>	<b>YES</b>	<b>YES</b>	<b>YES</b>
Call to Web Services	<ul style="list-style-type: none"> <li>• Main Data Web Service</li> <li>• Metadata Web Service</li> <li>• Discovery Web Service</li> </ul>	<b>YES</b>	<b>YES</b>	<b>YES</b>
Integrated Active Directory Authentication	<ul style="list-style-type: none"> <li>• Single sign on with underlying infrastructure</li> </ul>	<b>YES</b>	<b>YES</b>	<b>NO</b>
Forms Based Authentication	<ul style="list-style-type: none"> <li>• IFD for on-premise</li> </ul>	<b>YES</b>	<b>YES</b>	<b>NO</b>
Windows Live ID Authentication	<ul style="list-style-type: none"> <li>• Interactive authentication</li> <li>• Programmatic authentication</li> </ul>	<b>NO</b>	<b>NO</b>	<b>YES</b>
Client Side Mashups	<ul style="list-style-type: none"> <li>• IFrames</li> <li>• External Applications</li> </ul>	<b>YES</b>	<b>YES</b>	<b>YES</b>
Co-locate	<ul style="list-style-type: none"> <li>• Hosting ISV pages on CRM servers.</li> <li>• Hosting ISV code on CRM servers</li> </ul>	<b>YES</b>	<b>Partner Specific</b>	<b>NO</b>
.net plug ins	<ul style="list-style-type: none"> <li>• Synchronous call outs</li> <li>• Outbound integration triggers</li> </ul>	<b>YES</b>	<b>Partner Specific</b>	<b>NO</b>
Workflow	<ul style="list-style-type: none"> <li>• Declarative workflows using workflow tools in Web application</li> </ul>	<b>YES</b>	<b>YES</b>	<b>YES</b>
Custom Workflow Activities	<ul style="list-style-type: none"> <li>• .net Assemblies</li> <li>• External Web Service Calls</li> </ul>	<b>YES</b>	<b>Partner Specific</b>	<b>NO</b>
Offline SDK	<ul style="list-style-type: none"> <li>• Offline Outlook Client Programmability</li> </ul>	<b>YES</b>	<b>YES</b>	<b>Professional Plus Edition Only</b>

Using the above comparison chart ISVs should be looking for opportunities to bring their solution to the CRM Online environment in addition to targeting the other traditional deployment scenarios. By making your ISV solution available in all three environments you have opened the door for your solution to have a broader audience.

### 3.2. Packaging Solutions for Deployment

Packaging up your solution for deployment is an important part of the solution development lifecycle. This is particularly important if your solution is to be installed and managed by customer system administrators where your technical staff will not be involved in the installation. In this release of the platform there are a number of enhancements that continue to make it easier to perform customization tasks from within an installer. These improvements come in three key areas

Import/Export of customizations, APIs to manipulate Metadata and APIs to register workflow activities and plug-ins.

#### *Importing and Exporting Customizations*

Earlier we discussed several of the customization opportunities in the data model and the user interface areas. We are now going to explore how to get those to be deployable repeatedly. To accomplish this task the platform provides extensive customization import and export capabilities. Using this capability you can selectively export and then subsequently import all or a selective set of your customizations into a target system. New to Microsoft Dynamics CRM 4.0 is the ability to import and export security roles – this makes it easier to ship your product with customized roles.

#### *Using APIs to explore and manipulate Metadata*

All of the customizations that are performed to the platform make up the platform metadata. The metadata represents information about the data model and other platform specifics. The platform provides a robust API that exposes and allows you to query the metadata and use it as part of your custom processing. This capability can be useful during installation but can also be useful when building custom parts of your solution where leveraging the metadata would allow you to show the same label used by Dynamics CRM. It will even retrieve the label in the correct language for the user in multi-language environments. It also provides the ability to dynamically discover custom entities and custom attributes that have been added to the system. This discovery capability is particularly important if you're building solutions where you want to work with any of the provisioned custom entities. A good example is an import utility. It would be very cumbersome if you had to hard code all the entities, but using the metadata API you are able to interrogate and dynamically provide a list of custom entities.

New to Microsoft Dynamics CRM 4.0 is the ability to use the API to modify the platform metadata. This is an important addition to an already powerful API because it now allows you to customize the data model programmatically during execution of an installation routine, or even in response to a user requesting provisioning of a new system capability. For example, rather than importing a customization file, you could programmatically create custom entities or attributes. Probably the best example of this is when you're adding or altering one of the entities that come with the platform out of the box – by importing the customization you risk altering other attributes that are on the entity that the end user had customized already – by using the API to modify and add a custom attribute it's surgically added without risk to existing customizations. This provides greater flexibility and significantly reduces the risk of customization collision.

Another great use of the update capabilities of the metadata API is to use it to alter language specific customizations. Microsoft Dynamics CRM 4.0 now allows a single installation of the platform to have multiple active languages. They are configured to be active by installing language packs for the languages that will be used. Using the Metadata API an ISV could update labels and messages to provide support for various languages that are installed. The API makes it easy because you can either replace all languages or only provide additive information where you may, for instance, be adding support for a single additional language. Using the API, you could package up your language support to run as part of a larger install program or have a language pack install.

#### *Using APIs to Register Plug-ins and Workflow Activities*

In addition to being able to package up your metadata changes you can now also install your custom Plug-ins and Workflow Activities using the APIs. In the past registration of plug-in type customization required manual modification to the platform configuration files. Using the new APIs, ISVs are able to register plug-ins and provide any specific options such as what event is the trigger. During the registration process using the API, the binary for the plug-in is stored on the server. This allows for

efficient distribution of the plug-in binary to any of the platform servers when run in a multi sever configuration. Additionally, if the plug-in is configured for execution on an offline client, the binary will also be distributed to client machines. Using the API, plug-in registration can be completely automated and integrated into an ISVs installation package.

### 3.3. Wrapping up Deployment Essentials

The ability to package up solutions and deploy them into the end users' computing environment is important. Using support provided by the platform, such as the metadata API, ISVs are able to take complete control of how to integrate the platform installation process as part of an overall installation. Additionally, in the event the solution needs to be un-installed, these same capabilities help the ISV make sure their solution can be removed.

Microsoft Dynamics CRM 4.0 provides three distinct choices for where the server portion of the application may be deployed – On-premise, Partner Hosted or CRM Online. The Power of Choice expands ISVs opportunities to reach customers interested in on-demand services by Microsoft.

## 4. Operations Essentials

In this section of the paper we examine some of the aspects of the platform that help once your solution is deployed and in use by your customers. These are the platform capabilities and building blocks that allow you to tailor the platform and your solution to work in today's fast-paced business environment. Using these tools and techniques you will be able to allow customers to slice and dice data as well as monitor the platform and your solution to help ensure that it is available 24/7.

### 4.1. Integrating Business Intelligence

So far we have discussed how to tailor the platform to solve business problems by customizing, extending and automating business processes. Through those capabilities ISV solutions collect and manage data in a way that is natural for business users. 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, scorecards and other analysis to facilitate faster more accurate business decisions. ISVs using the building block capabilities of the platform such as reporting and analytics may integrate with other products like SharePoint to build customized solutions that target specific end user analysis needs.

#### *Building Reports – Online and Offline*

Microsoft Dynamics CRM 4.0 leverages SQL Server 2005 Reporting Services as the engine for producing reports. ISV developers can build custom reports to deploy with their solution using Microsoft Visual Studio 2005 and the report designer. End users can also use the new CRM 4.0 Reporting Wizard to create basic custom reports as needed using the client interface. The platform then manages these report definitions and allows them to be scheduled to create snapshots or to be used when the user is offline.

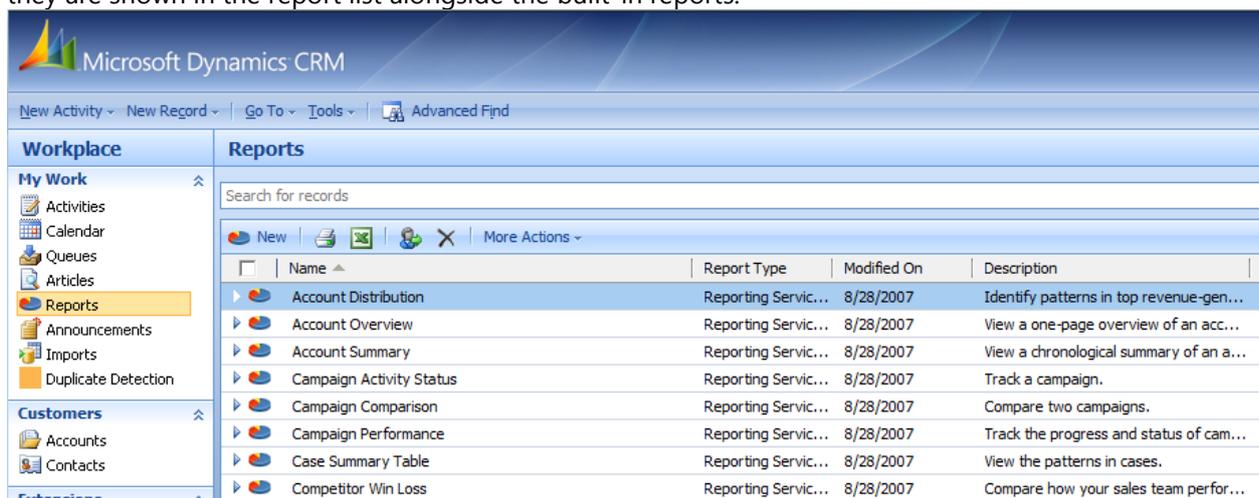
#### *Handling Report Security*

Earlier we talked about the importance of application security and how the platform 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 your solution. Microsoft Dynamics CRM

4.0 uses a concept called Filtered Views to provide that consistency in the security model. Filtered views are created and maintained by the platform for each built-in and custom entity defined in the data model of the solution. They incorporate the security roles, organization/business unit design and record ownership to enforce what data users have the ability to run reports against. Additionally, since reports are represented by a Report Entity the report itself can be controlled using the platform security model to determine who can see, run and modify the report.

#### *Out of the box reports*

Microsoft Dynamics CRM 4.0 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 ISVs 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 ISVs write and deploy custom reports as part of their solution they are shown in the report list alongside the built-in reports.

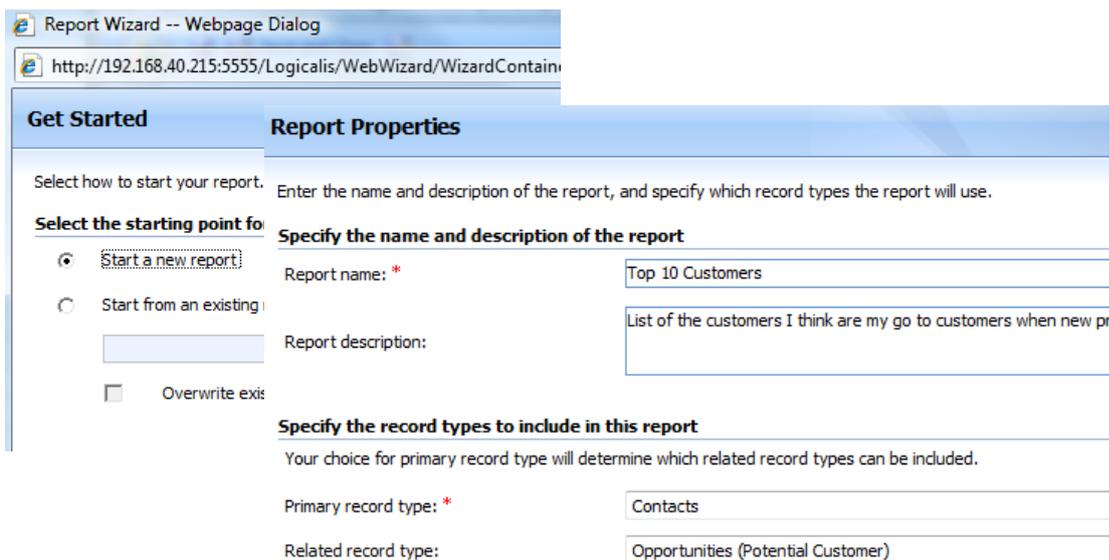


*Figure 11 – Custom reports are shown alongside built-in reports in the Reports list*

#### *New Report Wizard*

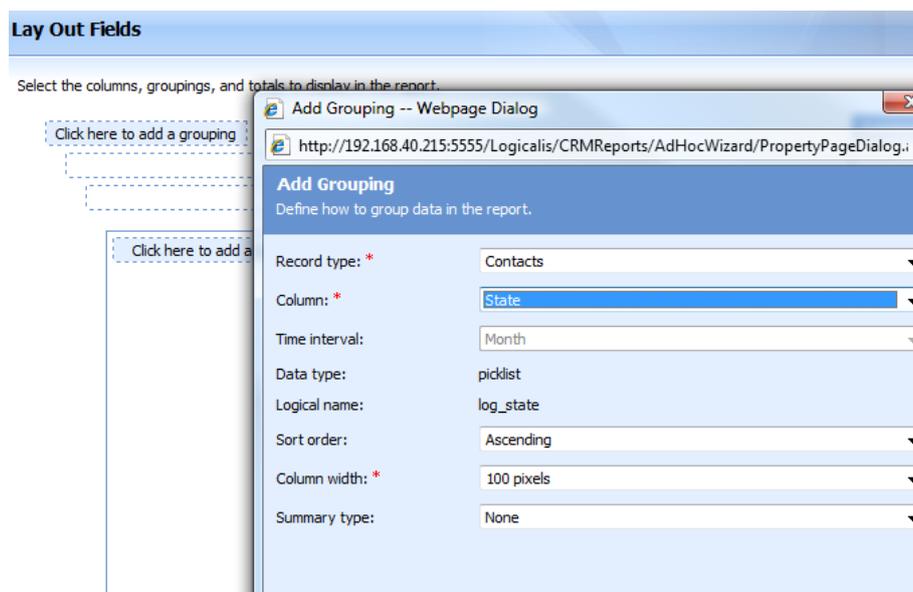
New to Microsoft Dynamics CRM 4.0 is the Report Wizard. The Report Wizard is geared to allow an end user to build a basic custom report on their own. This platform capability adds tremendous value to the ISV solution. This saves the ISV from having to build their own end user report writer; a non-trivial task.

The figure below shows the Report Wizard in action as it walks a user through creating a report. It's important to note that it can report on built-in entities as well as custom entities that the ISV may have defined as part of a solution.



*Figure 12 – The Dynamics CRM 4.0 Report Wizard walks the end-user through the steps of creating a report*

The user will use an interface similar to advanced find to define what data will qualify; in fact they can use a saved view. Next, the wizard will prompt the user for indicating any grouping of data and what fields should be shown on the report.



*Figure 13 – The Report Wizard gives the user the ability to choose columns as well the grouping order of those columns*

At this point the Report Wizard will save the report and it will look just like any other report in the system. That means they can schedule it, run it offline and treat it just like any other report in the system.

By allowing end users to create their own simple reports, customers will feel empowered to analyze more deeply the data in their systems on their own. This will leave the ISV more time to focus on more complex reporting and analytics.

### Out of the box reports

Businesses now more than ever depend on at-a-glance data that they can look at to get a high-level feel for how they are doing without having to invest hours sifting through detailed data. Dashboards, which are often combined with scorecard KPIs, have proven an excellent tool for articulating and monitoring specific business goals. ISVs can leverage Microsoft SharePoint in conjunction with the platform to provide a dashboard portal that integrates rolled up data from the platform in a view appropriate for business decision making. By customizing it as part of the ISV solution it can address specific solution needs that out-of-the-box reporting fails to address.

The following is an example of a dashboard in Microsoft SharePoint that integrates a scorecard for high level analysis, charts using a report viewer web part that allow for visual detail on specific metrics and finally the List web part that shows the top sales people.

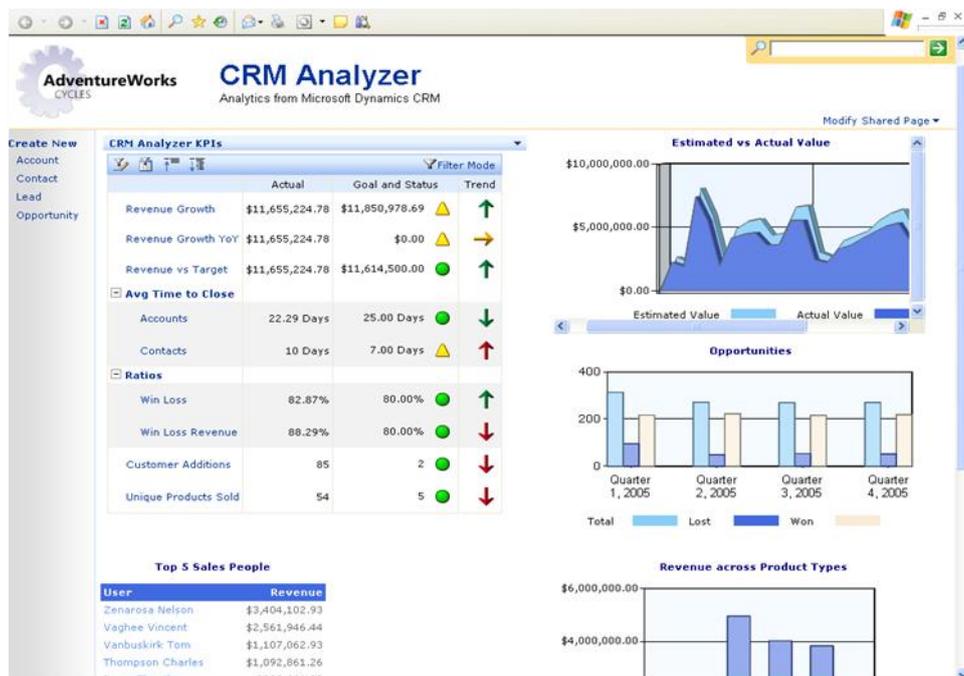


Figure 14 – Dynamics CRM together with Microsoft SharePoint provide an ideal solution for building dashboards and scorecards

### Analytics

So far all of the reporting we discussed involves detailed lists of data from the platform, or at-a-glance type data using dashboards for quick analysis. The other area of importance is the ability to do deep data mining and analysis to understand the patterns in the data and determine business impact. Microsoft Dynamics CRM 4.0 as a platform stores data in Microsoft SQL Server 2005 as relational data that is optimized for day to day use of that data. Deep data analysis typically involves pre-aggregated data to allow for fast analytical performance. The SQL Server 2005 platform includes SQL Server Analysis Services providing support for the concept of a Unified Dimension Model (UDM) or a Cube. Using the UDM concept, data is pre-aggregated and optimized for fast queries. The data included in the cube may include data from multiple databases in an organization.

Again, this is an area that ISVs can add onto their core platform solution providing their customers with deep analysis capabilities. To help jump start that process Microsoft has released Dynamics CRM Analytics Foundation.

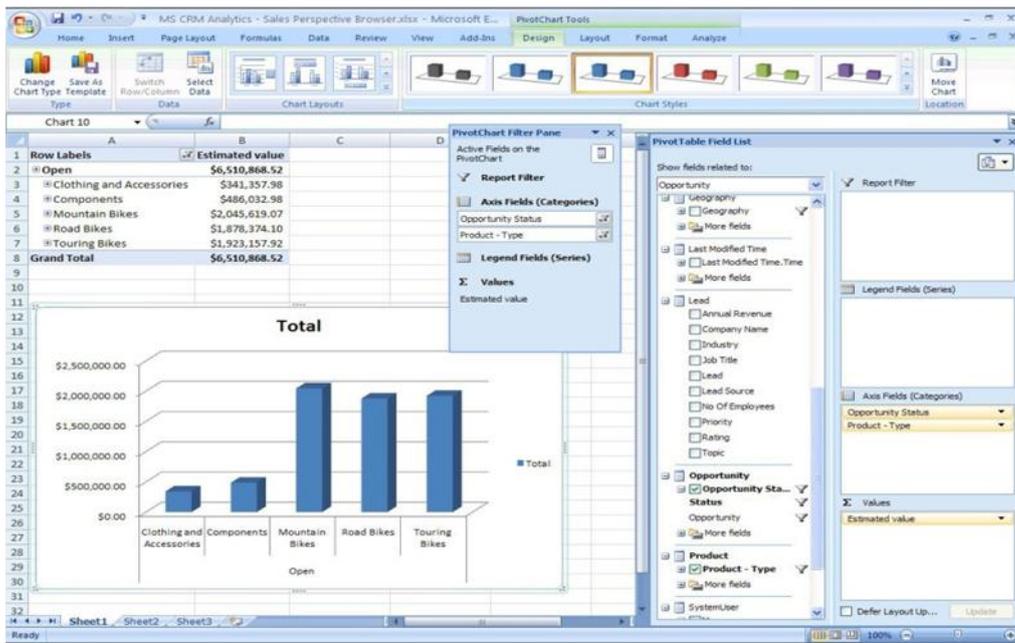


Figure 15 – Dynamics CRM Analytics Foundation provides customers with the ability to analyze the data in their CRM systems using tools such as Microsoft Excel

Dynamics CRM Analytics Foundation is not a solution in itself, but building blocks that allow ISVs to build deep analysis solutions. Included in Analytics Foundation is a pre-built set of cubes, dashboards, Excel reports and platform integrations showing how analytics can work in action. Using the foundation, ISVs have a great starting point to kick off their custom development efforts.

## 4.2. Managing Operations

Managing the day to day operation of the platform is vital when the application is mission critical to an organization and its employees. Using instrumentation built into the platform, operations can proactively monitor performance and other vital statistics. Proactive monitoring and automation helps ensure high availability, and substantially increases the productivity of support staff. The platform operation metrics can be viewed using though single purpose monitors for things like viewing Performance Counters, or using the integration with Microsoft System Center Operations Manager (SCOM) operations staff have a more complete view of the overall health of the system.

### *Performance Counters*

Performance metrics are implemented throughout the platform and recorded as standard Windows Server platform performance counters. The following are examples of some of the performance counters supported by the platform.



Figure 16 – Microsoft System Center Operations Manager monitors numerous Dynamics CRM performance counters

### Microsoft System Center Operations Manager

Using the integration with Microsoft System Center Operations Manager (SCOM), ISV solutions can provide customers an integrated operations monitoring solution. By providing a centralized view, operations staff can identify problems faster as well as use the automation capabilities of SCOM to respond to the problems encountered.

ISVs can further extend and provide support for their solution by integrating it as well into the SCOM framework.

## 5. What have we covered?

We have covered a lot of ground in the white paper. We've taken the ISV perspective to give you an idea of the capabilities of the Microsoft Dynamics CRM platform for developing complete solutions.

The solutions you will create will build on top of the robust core platform capabilities that Microsoft Dynamics CRM ships with, out of the box. Using these capabilities you can quickly build both horizontal and industry specific vertical solutions without having to custom build the basic foundation.

The features and capabilities added in Microsoft Dynamics CRM 4.0 continue the evolution of platform features that were available to ISVs in Microsoft Dynamics CRM 3.0. Many of the new features in Microsoft Dynamics CRM 4.0 will open the door for new types of applications and extensions to be built. Platform backward compatibility helps ensure that investments made by ISVs in the CRM 3.0 platform migrate forward to the 4.0 upgrade. We have focused on the key features of the platform that will allow ISVs to quickly build, deploy and manage an application built using the Microsoft Dynamics CRM 4.0 platform.

We hope we have accomplished our goal of sharing a high level understanding of the features and capabilities of the Microsoft Dynamics CRM 4.0 platform. With this basic understanding of the platform capabilities you can identify opportunities to within your own solutions to accelerate your development while providing your customers advanced capabilities. As you move past understanding the basics of the capabilities described in this document, additional resources such as the platform installation guide and the platform developer SDK provide you the detailed knowledge necessary for you to fully implement your solution.

## 6. Where to from here?

- ☑ Learn more about building your business on the Microsoft Dynamics platform
  - <http://www.innovateonmicrosoftdynamics.com>
- ☑ Join the Microsoft Partner Program, if you are not already a member:
  - <https://partner.microsoft.com/40009570>
- ☑ Use Microsoft Dynamics CRM within your own organization:
  - <https://partner.microsoft.com/global/program/programoverview/40029254>
- ☑ Learn more about the opportunities available to ISVs with Dynamics CRM
  - <https://partner.microsoft.com/global/productssolutions/dynamics/microsoftdynamicscrm/40043170>
- ☑ Download the software for development. This requires Microsoft Developer Network (MSDN) Universal membership. The MSDN license provides development and testing rights within your internal labs.
  - <https://partner.microsoft.com/global/productssolutions/business/businesscrm/40001496>
- ☑ Decide on the business model you would like to engage with Microsoft. Sign appropriate agreements specific to the engagement model (including ISV Royalty Agreement or Service Provider License Agreement).
  - <https://partner.microsoft.com/global/licensing/licensingprograms>
- ☑ Take advantage of the MSDN Microsoft Dynamics CRM Developer Center while building your solution
  - <http://msdn2.microsoft.com/en-us/dynamics/crm/default.aspx>
- ☑ Get your application tested by VeriTest
  - <http://www.veritest.com/lionbridge/en-US/services/outsourced-testing/product-certification-programs/microsoft/Microsoft-Dynamics-Test-for-ISV-Solutions.htm>
- ☑ Get your solution "Certified for Microsoft Dynamics"
  - <http://www.microsoft.com/dynamics/partners/cfmd.msp>
- ☑ Generate leads and find prospects for your finished solution
  - <https://partner.microsoft.com/global/40020720>

---

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](http://www.microsoft.com/dynamics)

**Microsoft**