



Microsoft CRM Customization

White Paper

Published: April 2003

For the latest information, please see <http://microsoft.com/crm>

Abstract

This paper describes how Microsoft® Business Solutions CRM provides a wealth of opportunities for solution providers and system integrators to create a highly customized and integrated application. Because it is easy to deploy, customize, integrate, and extend, Microsoft CRM helps to meet the needs of businesses.

Microsoft®

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 OR IMPLIED, IN THIS SUMMARY.

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.

© 2003 Microsoft Corporation. All rights reserved.

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

Microsoft, BizTalk, Jscript, Great Plains, Outlook, and Visual Basic are either registered trademarks or trademarks of Microsoft Corporation or Great Plains Software, Inc., in the United States and/or other countries. Great Plains Software, Inc., is a wholly-owned subsidiary of Microsoft Corporation. The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Contents

Introduction	1
Capturing Additional Information—Application Customization	2
Customizing schema	2
Customizing forms.....	2
Publishing changes to forms	3
Using the view editor	3
Customizing previews	4
Customizing e-mail templates	4
Replicating with the import/export feature.....	5
Customizing reports	5
Business Logic Customization.....	7
Using the Workflow Editor to create customized workflow processes.....	8
Microsoft CRM Extensibility	9
Microsoft CRM Integration	10
Integrating with third-party software	10
Using the BizTalk Mapper to customize integration maps	10
Managing changes to the channel	11
Using application integration components	11
Using pipeline components	12
Creating post-callouts.....	12
Using the OnChange events	13
Achieving application-level integration	13
Conclusion	14
Resources.....	15
Documents	15
Information and support	15

Introduction

Easy to deploy, customize, integrate, and extend, Microsoft Business Solutions CRM can be adapted to meet a wide range of organizational and industry-specific business needs. This paper describes how Microsoft CRM provides a wealth of opportunities for solution providers and system integrators to create a highly customized and integrated application. The Microsoft CRM metadata-driven architecture, declarative workflow engine, integrated setup, and tools have been designed to help increase the ease of configuration, customization, deployment, and upgrade.

As a true four-tiered Internet-architecture application, Microsoft CRM delivers a solution that can be scaled up or out, and that can be customized to a high degree. The application is partitioned into user interface (UI), application, XML Web services, and data tiers to support extremely flexible deployment options.

Unique to Microsoft CRM, the client UI has two distinct presentation components: Internet Explorer as the rich, Web-based user experience and Microsoft Outlook® as the Windows®-based experience. Both layers consume the same intermediate XML-based form definition language, which makes it possible for the system administrator, system integrator, value-added reseller, or IT manager to make UI customizations once and have them apply to both clients.

There are several levels of customization possible in Microsoft CRM—database schema, application (including customizing forms, views, and templates), and business processes. In the UI, forms, views, and previews can be modified to allow the business to achieve the level of specialization required. This level of customization can be superficial or require modifications to the database schema. Both types of customization can be accomplished easily with the System Customization area in the application, and with the Schema Manager tool in Deployment Manager of the Microsoft Management Console (MMC) snap-in.

Using the Workflow Manager tool, custom business processes can be created or changed quickly to fit the rapidly changing needs of an organization. Workflow rules can be as specific as creating a new opportunity for an incoming lead and assigning it to the salesperson of the relevant sales territory, or as generic as invoking a third-party Web service.

Workflow rules can be exported into a workflow XML file and all changes made to forms and schema can be exported to an XML file. The purpose of this export is to facilitate the implementation of vertical solutions in multiple businesses and to enable value-added resellers and system integrators to instantiate sets of common customizations rapidly.

Capturing Additional Information—Application Customization

Customizing schema

Before a custom attribute can be exposed to the Microsoft CRM user on a form, in a view, or in a preview, the attribute first must be added to the schema. Custom attribute used in workflow rules also must be added to the schema.

For example, a pet-store chain wants to be able to record “pet breed” on its Contact record for customers who are searching for a particular breed of pet. Before you can add the field to the contact form, you must extend the Contacts record by adding a “pet breed” attribute to the database schema.

To change the schema, you use the Deployment Manager, an MMC snap-in tool used for administering the schema, server, licenses and users. The Schema Manager found in this tool displays all of the objects in the application. Using the Schema Manager, you can view the attributes for an Object and create a new attribute for the chosen object. To add a custom attribute (“pet breed” in our example), select the type (picklist, string, Boolean, and so on) of the field you want to create, then set constraints. Finally, launch the Form Editor to add the new field to your form. The following section gives more specifics on using forms.

Customizing forms

Microsoft CRM comes with standardized forms that have been developed using “best practices” for CRM. A systems administrator can modify these forms to suit an organization’s need, whether to capture additional information or to modify the location of fields for ease of use.

Forms can be modified in the following ways:

- Add tabs
- Add sections to tabs
- Add fields to sections
- Change field labels, properties, values and add scripts
- Rearrange fields in a section.

An existing object attribute, or one that has been added to an object schema, can be added to a form by simply opening the Form Editor and selecting the attribute you would like to add (see Figure 1, below). The Form Editor adds the selected attribute and can be used to arrange the field on the form. You can make multiple changes to a form and publish the changes when you’re finished making the customizations.

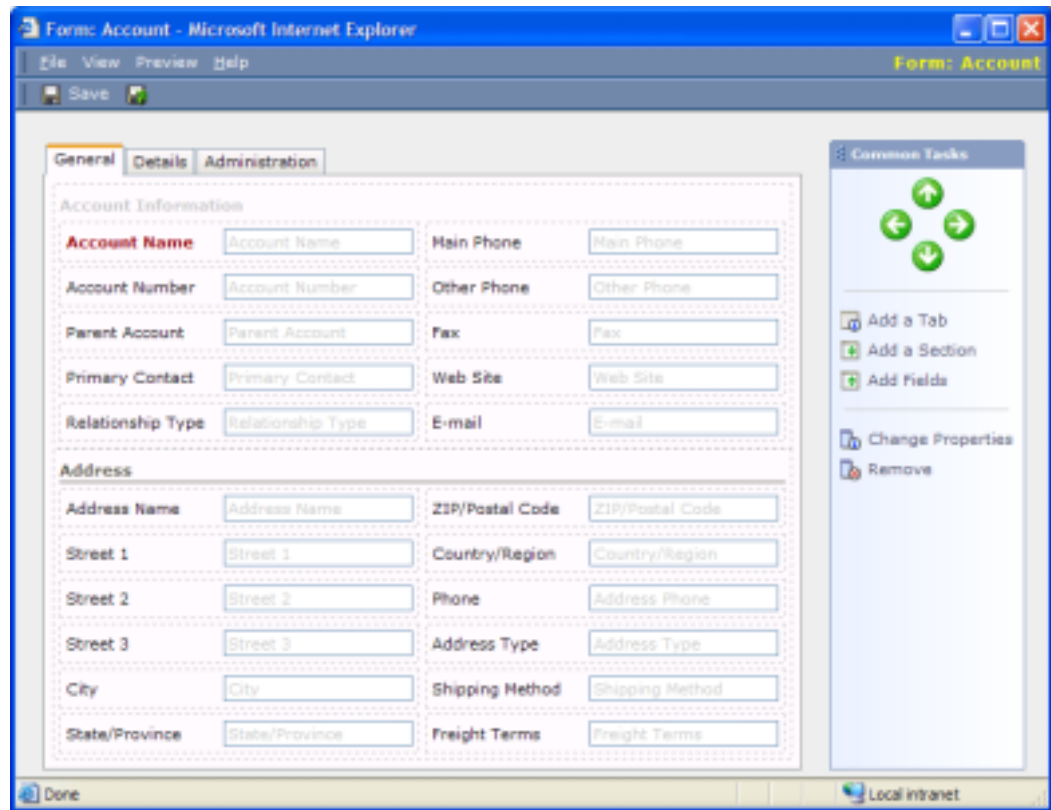


Figure 1. Screen capture of the form editor.

Publishing changes to forms

Once you're satisfied with the changes you've made, it's time to publish the revised form. From the Deployment Manager, select "Action;" next, select "All Tasks" and "Publish Customization." You can then select the objects you want to publish and the servers to which they should be published. The program will publish your customizations.

Using the view editor

Customizing views allows views to be created in order to view data in the way that makes most sense for a particular job function or company. For example, account managers may find it beneficial to view accounts based on annual revenue. Using the View Editor, you can filter accounts based on revenue ranges, for example, between \$0 and \$500,000, between \$500,001 and \$1M, and so on. Another example would be to create a view that shows accounts by territory of responsibility. The Microsoft CRM View Editor makes it easy to create views that filter on established requirements. Figure 2 shows the screen in the View Editor used to configure criteria for a view.

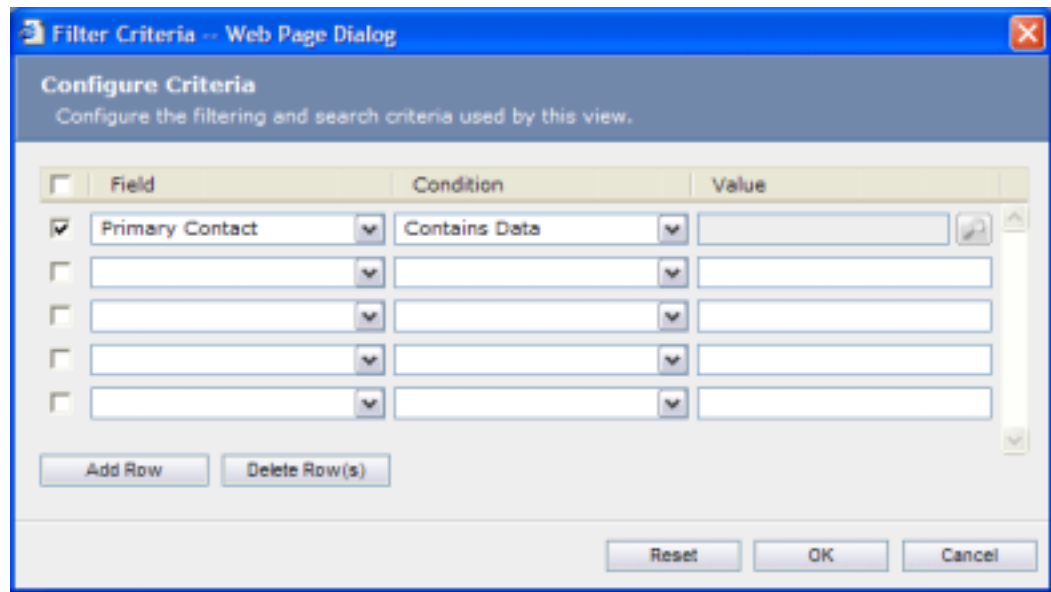


Figure 2. Screen capture of the Configure Filter Criteria Screen of the View Editor.

Users who want to see data other than what has been provided in the standard view can use Advanced Find on the Tools menu to create a customized query.

If you have a single Web server in your installation, once you have created a view, the view is automatically enabled. However, if you have multiple Web servers in your installation, you will need to publish your view.

Customizing previews

Microsoft CRM provides previews—in a way similar to the Preview Pane in Microsoft Outlook—letting users view selected information on a form without opening the form, merely by clicking on the carat next to the displayed item.

To customize a preview, use the “Customize Preview” editor. A preview may contain up to ten fields and does not contain tabs. You can add or remove fields and also move them up, down, left and right.

Customizing e-mail templates

E-mail templates allow users to customize e-mail correspondence with account contacts. Microsoft CRM ships with a number of pre-configured templates. Because they're formatted and edited using HTML, it's easy to customize and create great-looking e-mail correspondence that makes an impact.

Microsoft CRM provides an e-mail template editor (see Figure 3, below) that lets users adjust the format and choose which objects the template will use. Inserting data slugs where they are needed in the e-mail lets users merge information from the Microsoft CRM database, such as account contact name and other unique information and, in this way, personalizes e-mail messages, saving users time copying or cutting and pasting information.

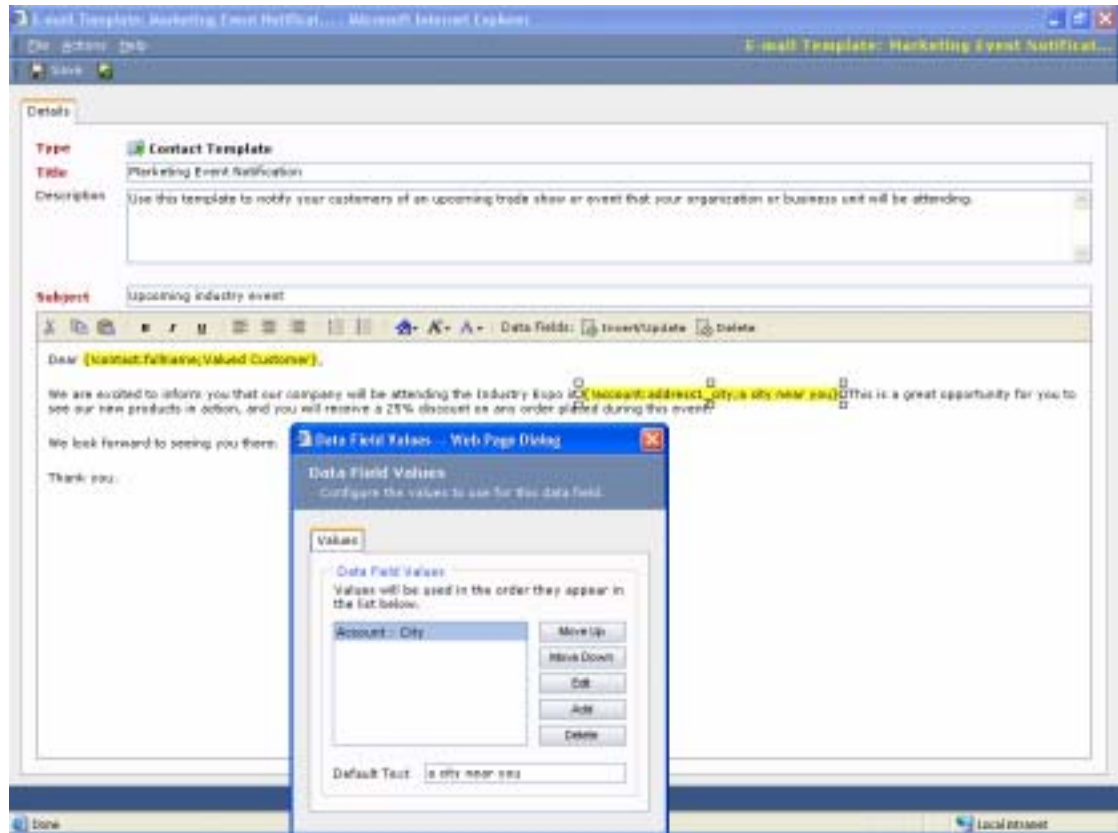


Figure 3. Screen capture of the E-mail Template Editor.

Replicating with the import/export feature

All changes made to the forms, schema, and templates can be exported to a single XML file. Likewise, workflow rules can be exported to an XML file. This powerful feature allows system integrators and value-added resellers to create a set of customized schema, forms, views, and templates and easily export them to another server in a single XML file. Customization work done for a particular vertical market one time can be replicated and offered to multiple customers. To use this feature, the customer must be using the same version of Microsoft CRM as the reseller. The Import/Export feature can be accessed through the Microsoft CRM Deployment Manager.

Customizing reports

The Microsoft CRM application comes with a robust reporting component from Crystal Decisions that makes it easy to access summarized information from the system. Although Microsoft CRM provides more than 100 standard reports covering every component of the product—from account hierarchy reports to pipeline charts to case management effectiveness reports, customers may wish to add new reports or customize existing ones.

New reports can be created to meet a business's unique needs. For example, a business may want to provide its sales force with the opportunity to provide an instant quote. A quote detail report could be created and customized for this purpose, providing the salesperson the opportunity to make the sale and close the deal on the spot.

Standard reports also can be modified to include new fields for reporting information unique to an organization or industry. A business may wish to display more information on an account report, for example, or include additional fields. Reports can be customized using Crystal Reports.

Although this paper summarizes the basics of Microsoft CRM report customization capability, another document, *Customizing Reports for Microsoft CRM Project Guide*, provides detailed information on upgrading to Crystal Reports Enterprise Edition and customizing reports.

Crystal Reports is the design environment for creating and modifying the report templates that are published to Crystal Enterprise. Built-in report experts guide users, step by step, through building reports and completing common reporting tasks. Formulas, cross-tabs, sub-reports, and conditional formatting help users make sense of data and uncover important relationships.

To be able to create new or to modify existing Microsoft CRM reports and publish them to Crystal Enterprise, you need, first, to install a retail version of Crystal Reports and then subsequently install the Crystal Reports Enhancement for Microsoft CRM. The following describes the available versions of Crystal Reports as they relate to Microsoft CRM:

- **Crystal Reports Professional Version 8.5¹** (purchased separately)—This is the design environment. The component used to customize reports is the Report Designer.
- **Crystal Reports Enhancement for Microsoft CRM** (provided with Microsoft CRM)—This is a set of Microsoft CRM-specific enhancements to Crystal Reports, including:
 - The ability to publish reports directly to Enterprise folders
 - Crystal Driver for Microsoft—CRM provides access to Microsoft CRM data to use in reports
 - A new chart type, called a funnel chart. Funnel charts are often used to represent stages in a sales process and show the amount of potential revenue for each stage. This type of chart can be useful also in identifying potential problem areas in an organization's sales processes. A funnel chart is similar to a stacked bar chart and represents 100% of the summary values for the groups included in the chart.
- **Crystal Reports 8.5 for Microsoft CRM** (purchased separately) — contains the same functionality as Crystal Reports 8.5, except that it only can be used to connect to the Microsoft CRM database.

¹ Although newer versions of Crystal Reports are currently available, only version 8.5 of Crystal Reports may be used to customize reports for Microsoft CRM. In addition, only the Professional and Developer editions of the retail product may be used (the Standard edition is not supported).

Business Logic Customization

A key component of Microsoft CRM is its ability to create customized workflow processes. The Workflow tool allows activity to happen without direct interaction by the users of the system, creating workflow processes by applying rules set up by the business. For example, every time a customer places a new order, the system automatically sends an e-mail, acknowledging the purchase and thanking the customer. Advantages of automating such tasks include helping to save time and money and helping to improve customer relations.

For administrators, Microsoft CRM provides a Workflow tool that enables an organization to create processes that have a series of steps or actions (for example, the process of making a purchase) initiated by the occurrence of some event (for example, the creation of an order). Within each step, the workflow can evaluate a “condition,” price or time (for example, “if” the order is greater than \$10,000) and activate a series of actions based upon evaluation of that condition.

Following are some of the types of scenarios for which workflow customization is beneficial.

Lead assignment and routing

A company has several regional offices in different parts of the United States. The company Web site collects customer information on a “Send me more information” page. This page generates leads into Microsoft CRM. A lead is created and, based on a customer’s geographic location, that lead is routed to the appropriate sales territory manager.

Serial sales process

Many companies have sales processes which they like to follow so they can better manage sales opportunities from start through close. The use of standardized procedures helps define and add consistency to the flow of salespeople’s activities because there are sets of activities for each step in the sales process. In addition, by following a sales process, management can better assess the potential value of the company’s sales pipeline because, at each milestone or step in the sales process, management can estimate the likelihood of continued success, or of a new opportunity, or of an eventual sale.

When a new sales opportunity presents itself, a company may want the opportunity to move through a standardized process consisting of multiple stages (for example, from initial customer contact, through information gathering and negotiation, to closure of the sale) before the deal has been deemed “won” or “lost.” At each stage in the process, a series of activities can be assigned to salespersons, using workflow (for example, creating a proposal, plus meeting with a potential customer); until those activities have been completed, the sale cannot progress.

Order fulfillment notification

A salesperson has completed the steps to turn a sales opportunity into an order. The created order is submitted to the back office system where the order will be processed and the items will be shipped. Shipping prompts an update to a line item, indicating status changes. The CRM administrator has set up a workflow rule so that every time a line item status changes, an e-mail can be sent, notifying the customer that the item has been shipped.

Using the Workflow Editor to create customized workflow processes

Microsoft CRM provides a Workflow Manager and Editor (see Figure 4, below) for creating custom workflow processes. Creating rules and conditions is accomplished simply by selecting from the structures and conditions provided in the editor.

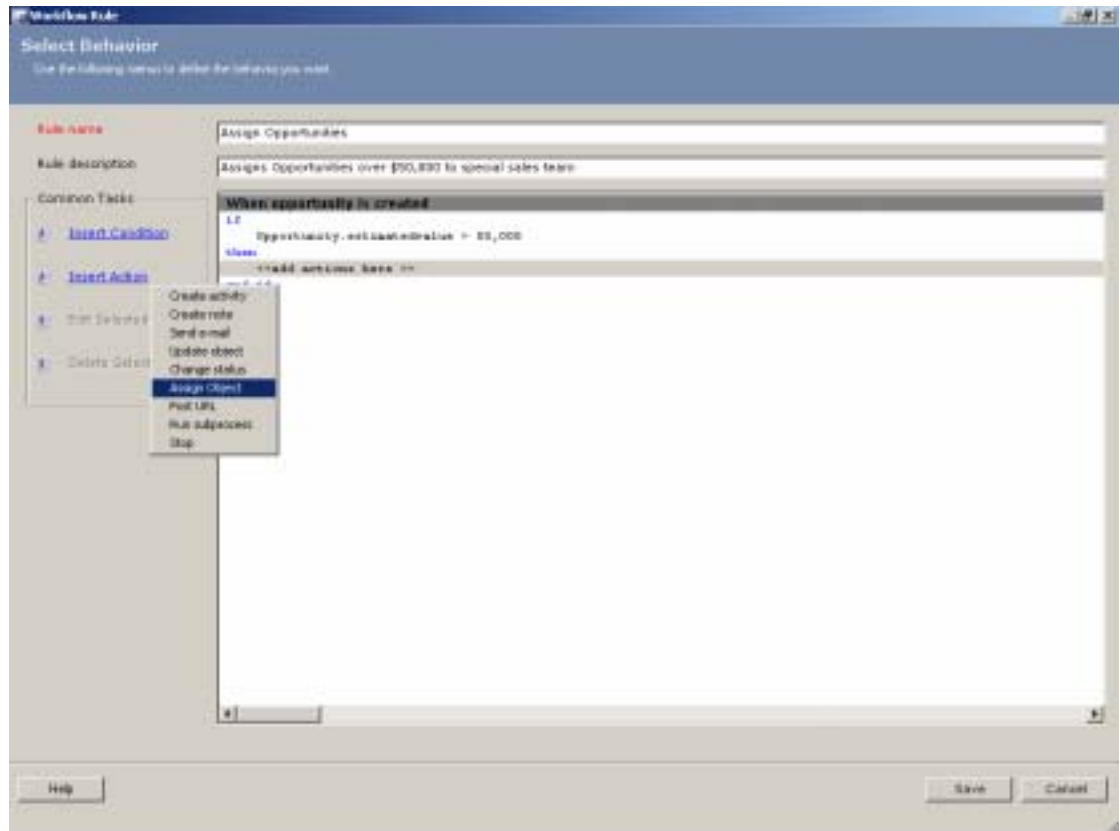


Figure 4. Screen capture of the Workflow Rule Editor

Once a workflow rule has been created, you must activate the rule and, having done so, the rule will operate whenever the related condition has been met.

Microsoft CRM Extensibility

Although Microsoft CRM handles common customizations with visual tools, extending Microsoft CRM with code allows an almost limitless means for customization. Developers can extend the functionality of Microsoft CRM and interface other systems with Microsoft CRM, and they can write modules invoking specific actions to achieve various results.

Developers can add menus to the application that, in turn, sends data to custom code modules, such as integrating a credit check to a third party into the Account form, or showing shipping status on an order directly from the shipping company. Internal systems and code can be added to provide a richer experience for the entire organization.

The [Microsoft Business Solutions CRM Software Development Kit](#) (Microsoft CRM SDK) contains resources designed to help developers build powerful vertical applications that extend the functionality of Microsoft CRM. The Microsoft CRM SDK includes documentation that provides a wide range of instructive and practical information. In addition to a comprehensive class library reference, the documentation offers conceptual overviews, integration details, database schemas, and sample code.

Microsoft CRM Integration

One of the most important aspects of a line of business application is the exchange of data with other applications. A white paper entitled [Microsoft CRM Integration](#) details the functionality and design of features that enable Microsoft CRM to integrate with other Microsoft Business Solutions applications, as well as with third-party applications.

Integrating with third-party software

Using the Microsoft CRM SDK, Microsoft CRM BizTalk adapter, and BizTalk Server 2002 Partner Edition, developers can build integrations to third-party applications. Microsoft CRM provides the necessary building blocks for integrating with third-party software using custom integration maps. To build integrations between Microsoft CRM and a third-party application, the Microsoft CRM AIC (Application Integration Components) is the starting point.

Using the BizTalk Mapper to customize integration maps

The integration between Microsoft CRM and Microsoft Business Solutions—Microsoft Great Plains® and Microsoft Business Solutions—Solomon can be customized using BizTalk Mapper. This tool allows customization of existing field mappings, as well as the addition of new field mappings.

Capability for field mapping customization can provide powerful time-saving benefits. For example, a company may customize its Microsoft CRM order form, capturing additional information unique to its business. That information can be mapped and transferred to Great Plains, without needing to re-key customer and other order information.

The BizTalk Mapper has been designed to help support complex structural transformations from records and fields in the source specification tree (either Microsoft CRM or Great Plains) to records and fields in the destination specification tree. Because of this support capacity, BizTalk Mapper is an ideal tool for customizing the integration between Microsoft CRM and Great Plains. This tool allows a developer to write logic describing how data in the source specification should be transformed to data in the destination specification.

The core mechanism for performing these transformations is called a *functoid*.

A functoid is a predefined formula that takes in arguments and then performs specific calculations. For example, BizTalk Mapper provides an Addition functoid that takes the values of designated fields on the source side and then outputs the result to a field on the destination side.

Because not all possible data transformations can be done using the predefined functoids that Microsoft BizTalk Server 2002 provides, there is also a way to create custom functoids. To use custom functoids, simply use the Scripting functoid on the Advanced palette. Custom functoids can be written either in Microsoft Visual Basic® Scripting Edition (VBScript) or Microsoft JScript® development software.

The following VBScript code helps produce the mapping between a Microsoft CRM account address and a customer address in Microsoft Business Solutions—Great Plains:

```
1. Function SetActionValue( Action, EmptyValues, SourceSystemID,
SourceEntityID, MessageId, Required )
2. Dim retValue, sExistingLink
3. If LCase(Action) = "ignore" Then
4. retValue = "ignore"
```

```

5. Else
6. sExistingLink = MSCRMObj.GetDestinationLink( SourceSystemID,
gSourceEntityType, SourceEntityID, MessageId, gChannelName,
gDestSystem, gDestObjectType, Required )
7. If trim(sExistingLink) = "" Then
8. sExistingLink = "false"
9. Else
10. sExistingLink = "true"
11. End If
12. End Function

```

The first line of code establishes the function name and the parameters that it is taking in. In this case, the parameters come from the header of the XML message. The second line declares two variables that are going to be used in the function. The third line of code checks to determine if the action is "ignore;" if it is, then the code stops here. In line 6, the code tries the other branch because the action is not set to ignore.

In working with line 6 of the code, developers can use BizTalk Server to leverage the code that already exists within the Microsoft CRM Integration platform. The object called **MSCRMObj** actually has been instantiated previously and then reused here. The method **GetDestinationLink** is actually in a method in the dynamic link library (DLL) called XSLUtility. The remaining code lines (7-11) decipher the return value of the method and set the variable of **sExistingLink** to true or false, based on the return value.

Once you make modifications to a map using BizTalk Mapper, it is important to verify that the new map is syntactically correct. To do so, you can use the compile feature, which will take the map that the developer has created and enforce the structural rules and transformations that are in the mapping grid.

The BizTalk Mapper tool enables developers to write highly customized transformations between Microsoft CRM and Microsoft Business Solutions–Great Plains.

Managing changes to the channel

Maps that define data transformation can reside in two places:

- A local hard drive
- Web Distributed Authoring and Versioning (WebDAV).

By default, all maps are published to WebDAV so that BizTalk Messaging Manager can use them. If, however, you want to customize maps and then test them before actually publishing them to WebDAV, you can save the customized map to the local hard drive. To customize and test a map, go into BizTalk Mapper, retrieve a map from WebDAV, and make all of the desired customizations. Then save the map to the local hard drive and test the map there. After testing, you can save the map to WebDAV, so it can be used collaboratively.

Using application integration components

Microsoft CRM Integration uses AIC to deliver data into both Microsoft CRM and Great Plains. The AIC allows for an integration that can directly bind to BizTalk Messaging Services and activate and deliver data to custom components. With Microsoft CRM, the AIC is talking directly to the platform layer of Microsoft CRM. With Microsoft Business Solutions–Great Plains, the AIC is talking to a set of customer objects that mimic the business logic present in the user interface of those applications, thus maintaining all financial restrictions and logic.

Using pipeline components

Another integral part of Microsoft CRM Integration is its use of pipeline components. Once the data has been transformed by the channel, it calls a pipeline component; this component then communicates with the AIC. In short, it is the pipeline component, not the BizTalk Server channel, that communicates directly with the AIC.

The pipeline component has three major responsibilities inside the system:

1. **Responsibility for linking messages together:** The pipeline component, as detailed below, is responsible for linking messages together. That is, after a message has been sent to the AIC, the linking mechanism is responsible for persisting the fact that a certain account in Microsoft CRM has been linked to a certain customer in Microsoft Business Solutions—Great Plains.
2. **Responsibility for status messages:** The pipeline component is also responsible for handling status messages. Once the AIC has completed its task, it will send back a status message on whether or not the operation succeeded. The pipeline component is responsible for taking the status message and sending it to the appropriate channel so that it goes back to the system from where the call originated.
3. **Responsibility for posting messages:** The pipeline component is also responsible for posting messages that it receives from the channel into the AIC.

Creating post-callouts

Microsoft CRM exposes a very simple extension mechanism for implementing custom, platform-based business logic—a COM-based "callout" model. With this model, application developers are not limited to creating custom business logic only in the application or through workflow processes; they also can construct business logic that runs in the context of a call, but in response to a particular event.

This extension mechanism exposes a callout interface based on a simple pipeline model. The model allows for "post" callouts for the most basic data persistence operations: **Create**, **Update** and **Delete**. The platform metadata stores information about each entity. This information can be used to track the list of callout handlers, the class names, and whether a given handler is required for an action. For example, the **Account** object may have several registered handlers. These handlers are stored in call order (basically, a priority order). When an action occurs, for example **Account::Create**, the platform checks the metadata for registered event handlers, storing the information in the metadata and caching it in the platform server. If a handler is registered, the platform creates the handler object and calls the correct handler method, passing the original XML document.

There are three tables in Microsoft CRM that must be updated to register your callouts:

- The **Subscriber** table contains a record registering the global unique identifiers (GUIDs) of each class that implement the **ICRMCallout** interface.
- The **EntityEventSubscribers** table contains a record for each entity/event pair you are registering. This record also contains a bit indicating if the callout is enabled or disabled.
- The **Event** table contains an **EventMask** column which must be set for the metadata to recognize that events have been registered for a particular entity

Using the OnChange events

In Microsoft CRM, a customizer can add to a field on a form a script that will be triggered when the data in the field changes. This customization could be used for validation purposes or to fill in another field, for example, to fill in the state when the user enters a zip code.

Achieving application-level integration

It's possible to integrate other applications and business practices into Microsoft CRM; for example, you can add a button that will perform a lookup in an external database. The application integration section of the Microsoft CRM SDK covers this feature in detail.

Any integration into the application must participate in security. That is, any server or client application calling into the application programming interface (API) must have user context in the same domain. Before integrating applications at the API level, it's important to understand the Security Model and the Object Model in Microsoft CRM. The Microsoft CRM [SDK](#) provides the necessary background information.

Conclusion

As you can see, Microsoft CRM can be easy to deploy, customize, integrate, and extend, providing a wealth of opportunities for solution providers and system integrators to create a highly customized and integrated application for businesses. Its powerful import/export feature allows system integrators and value-added resellers to create a set of customized schema, forms, views, and templates and easily export them to another server in a single XML file. A list of additional information on customizing, integrating, and extending Microsoft CRM can be found in the section that follows.

Resources

Documents

[*Customizing Reports for Microsoft CRM Project Guide*](#)

Information and support

Microsoft offers many options for obtaining information and support.

MSDN

The MSDN® Web site is the source for the [*Microsoft CRM SDK*](#), technical articles and case studies on integration and customization, code samples, and other downloads.

<http://msdn.microsoft.com/>

TechNet

Microsoft TechNet is the central information and community resource for IT professionals. Here you will find the latest CRM articles and information on security, deployment, and operational best practices and tools.

<http://www.microsoft.com/technet/>

PartnerSource

These Web sites provide the latest product information and other valuable resources.

<http://www.greatplains.com/partnersource/mscrm/default.asp>

<http://members.microsoft.com/partner/default.aspx>

ISV Partner Support

In addition to support provided to MSDN subscribers, Microsoft Business Solutions (MBS) Solution, developers receive support from the MBS Tools Support team by phone and eSupport.

<http://www.greatplains.com/partnersource/support/>

<http://members.microsoft.com/partner/default.aspx>