Microsoft®
Customer
Relationship
Management

Microsoft CRM Integration

White Paper

By Eytan Seidman
Microsoft Corporation
Published: December 2002

Abstract

This paper outlines the functionality and design of features that enable Microsoft® Business Solutions Customer Relationship Management to integrate with other Microsoft Business Solutions applications, as well as with third-party applications. This functionality helps businesses synchronize data across disparate systems and customize how this data is synchronized.

Contents

Acknowledgements

Introduction

Microsoft® Business Solutions Customer Relationship Management (Microsoft CRM) software is a line-of-business application designed to help revolutionize the productivity and total cost of ownership of CRM applications for the mid-market.[1] Built to deliver an outstanding out-of-the-box experience for customers, Microsoft CRM also gives value-added resellers and independent software vendors a platform for extending the solution.

Microsoft CRM Integration is a component of Microsoft CRM and a complete solution for integrating with Microsoft Business Solutions applications, including Microsoft Business Solutions—Great Plains Release 7.0 , Microsoft Business Solutions—Solomon IV Release 5.0, Microsoft Business Solutions—Navision, and Microsoft Business Solutions—Axapta.[2]

This document covers design aspects and features that will help you:

- Understand how Microsoft CRM Integration uses Microsoft BizTalk® Server 2002

- Understand the platform upon which Microsoft CRM Integration is built

- Explore and change settings with the Settings and Administration tool

- Customize field mappings using Microsoft BizTalk Mapper

- Learn about the end-user experience on objects that are integrated.

How Microsoft CRM Integration Works

The primary service of Microsoft CRM Integration is to provide ongoing synchronization of data between Microsoft CRM and Microsoft Business Solutions—Great Plains and Microsoft Business Solutions—Solomon. Microsoft CRM Integration also offers the following additional services:

- A Settings and Administration tool that allows a user to configure settings related to what data should be moved between the systems, and that also details particular business-logic settings.

- Microsoft BizTalk Server 2002, which allows for rich customization of how data is transformed between the two systems.

The Settings and Administration tool offers more functionality than in previous versions of the integration and does so in a way that is easier for the user. Furthermore, BizTalk Server 2002 provides a robust platform for integration with Microsoft Business Solutions applications and to build out integrations with other third-party applications.

---

[1] Mid-market is defined here as companies with between 25 and 1,000 employees.

[2] Microsoft Customer Relationship Management Integration for Great Plains will be released very close to the same time as the release of Microsoft CRM . Integration with Solomon, Navision, and Axapta will follow.

Core Integration Service and Customization

Microsoft CRM Integration architecture

Microsoft CRM Integration is built on top of Microsoft BizTalk Server 2002 Partner Edition. Microsoft CRM Integration supports the synchronization of data between business applications by providing the following:

- Components that can publish data out of the line-of-business application based on parameters, such as whether the entity has been updated

- A mechanism that provides for guaranteed delivery of messages between the two systems

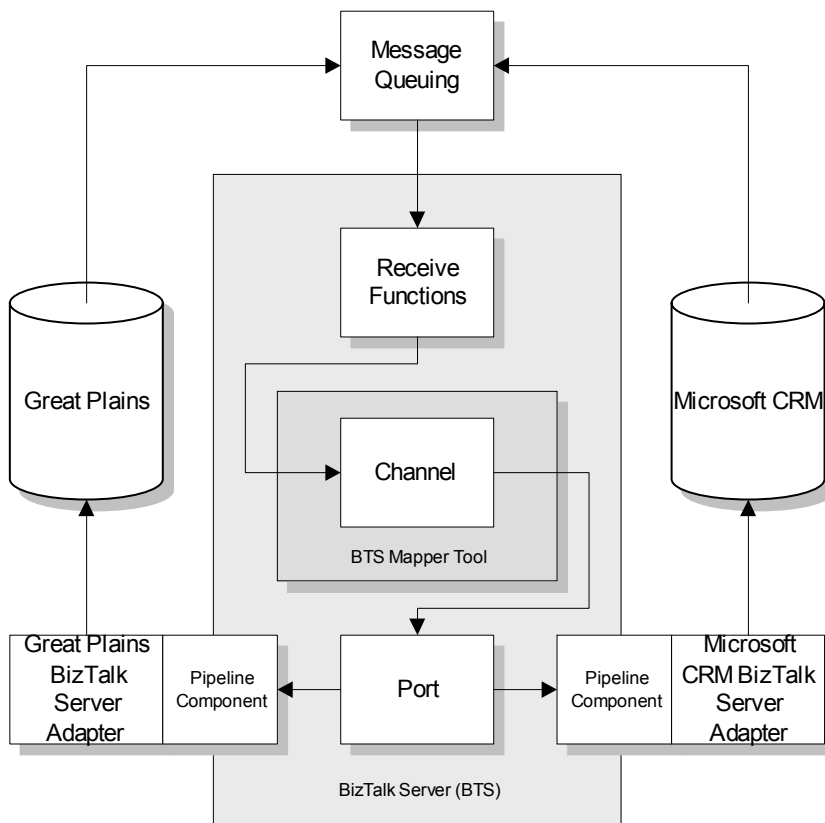- A mechanism for transforming data and adding necessary business logic.

Figure 1. A High-level view of Microsoft CRM Integration architecture.

Publishing data to a message queue

The process begins when an event occurs in one of the two systems. For example, after an account has been created, Microsoft CRM uses its metadata to know the event should be

recognized and transmits that data to a specified location. Microsoft CRM Integration then publishes the data to the Microsoft Message Queue (MSMQ); however, the data could be published to other systems, such as the file system. Microsoft CRM only publishes those fields that have been changed. That is, if a user updates only the account name on an account, then only the account name will be sent. The data is sent out as a message in Extensible Markup Language (XML) format, specifically in XML-Data Reduced (XDR) format.[3]

To publish data out of Microsoft Business Solutions—Great Plains, a slightly different mechanism is used. The mechanism will be referred to broadly as Document Exchange (DX). DX is composed of Microsoft SQL Server™ triggers and Microsoft SQL Server Data Transformation Services (DTS) packages. To understand how data moves out of Microsoft Business Solutions-- Great Plains, consider the following description of a single update to a customer record:

1. A user updates a customer record.

2. A Microsoft SQL Server trigger takes the data from the customer record and inserts the key data into the shadow table.

3. A DX package runs according to schedule and moves the key data from the shadow table to the work table using a stored procedure.

4. A DX package runs another stored procedure that looks for all of the necessary data that needs to be published.

5. Once all data that needs to be published has been found, it is placed by DX onto a queue—specifically, in this case, to MSMQ.

In summary, data publishing occurs quite differently in Microsoft Business Solutions—Great Plains than in Microsoft CRM because of the differing architectures. The result, however, is the same: Data is published to a message queue.

Receive functions

Once the data has been dropped off onto the queue, Microsoft CRM itself no longer plays a role in the integration. That is, Microsoft CRM considers the action as "fire and forget." Microsoft CRM Integration configures a BizTalk Server receive function upon setup. This receive function is used to pick up data from various systems. In the case of Microsoft CRM Integration, it will pick up data from the queue from which Microsoft CRM has published.

Microsoft CRM Integration preprocessor

After picking up the data off the queue, the Microsoft CRM Integration preprocessor analyzes it. The preprocessor determines whether it needs more data from Microsoft CRM and, if so, queries for the necessary records. An example would be the creation of a new account in Microsoft CRM. The account is sent out to MSMQ, where the preprocessor picks up and then examines the data. The preprocessor determines that it needs all of the addresses related to that account and

---

[3] BizTalk Server 2002 does not yet support Extensible Structure Definition (XSD) format. Once BizTalk Server supports XSD, Microsoft CRM Application Integration will as well.

queries back into Microsoft CRM for all of the addresses. Once all information has been obtained by the system, the message is sent by BizTalk Messaging Services to all appropriate channels. (For definitions of BizTalk Server Channels, see Appendix A.) Each channel determines what inbound document it has received and, subsequently, which map to apply and what outbound document it will transform into. Once the transformed outbound document has been created, it is then sent to the appropriate application integration component (AIC).

Channels and the BizTalk Mapper tool

BizTalk Channels are the heart of Microsoft CRM Integration. Channels are crucial to the integration, because it is within the channel that much of the transformation occurs. A channel is defined by three pieces of information: one inbound document (an XML file), one outbound document (also an XML file), and a XSL file map that dictates how the inbound document is transformed into the outbound document. By using the BizTalk Mapper, a developer can make modifications to existing mappings or introduce new mappings to fields that are not mapped out of the box. Creating new mappings between custom fields in Microsoft CRM and user-defined fields in Microsoft Business Solutions—Great Plains is supported without any caveats. Making additional changes to existing modifications is allowed, but with extra support from Microsoft Business Solutions to ensure that the integrity of the system will not be compromised by making such modifications.

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 Microsoft Business Solutions—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 Microsoft Business Solutions—Great Plains. It allows a developer to write logic easily about 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.

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.

In summary, 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.

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, which communicates directly with the AIC.

The pipeline component has three major responsibilities inside of 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. (For details, see the "End User Experience" section, p.9)

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.

Using application integration components

Microsoft CRM Integration uses AICs to deliver data into both Microsoft CRM and Microsoft Business Solutions—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

For developers who want to build integrations between Microsoft CRM and a third-party application, the Microsoft CRM AIC is the starting point.

Microsoft CRM Integration linking system

The pipeline component is responsible for the Microsoft CRM Integration linking system, which writes to a database located on the Integration server where Microsoft CRM Integration has been installed. This database stores linking information about the message received. For example, assume that the integration system received a "create account" message from Microsoft CRM. This message will create a customer in Microsoft Business Solutions—Great Plains. Before sending the data to the financial system, the tracking system will write the Primary Key (PK) of the Microsoft CRM account into this database. Once the customer has been created in the financial system, the customer components that the AIC called into will return a status message with the details of the customer that it creates. Most importantly, the AIC will pass the PK of the customer, and then insert a row set into the table, as shown in Figure 2.

OType	OID_CRM	OID_Microsoft Business Solutions—Great Plains
Account	{98197F09-3E7A-414E-9105-5AC2F438477F}	FOO123456
Contact	{A8A05D04-2068-45EE-9766-5E89950B8176}	BAR123456
...	...	...

Figure 2: A table of key information for Microsoft CRM and Microsoft Business Solutions—Great Plains. Note that the Microsoft Business Solutions—Great Plains key is actually a string entered by the user and is different from the system-generated, globally unique identifier (GUID) in Microsoft CRM.

If an "update request" comes from Microsoft Business Solutions—Great Plains asking to update the record with a PK of BAR123456, the tracking subsystem will have all of the information it needs and will not have to rely on any other system for information. Furthermore, with this design, Microsoft CRM does not have to store any data about the system that it has integrated within its own database.

Microsoft CRM Integration – Settings and Administration Tool

Settings and Administration tool

The Settings and Administration tool is a control panel designed to provide the administrator of Microsoft CRM Integration an area in which to manage the integration. The Settings and Administration tool has two main goals:

- Allow the business to configure how and what data moves between Microsoft CRM and Microsoft Business Solutions—Great Plains

- Easily map those fields that would be otherwise very difficult to map.

To enhance ease of use, the Settings and Administration tool has a user interface that is consistent with the Microsoft CRM user interface.
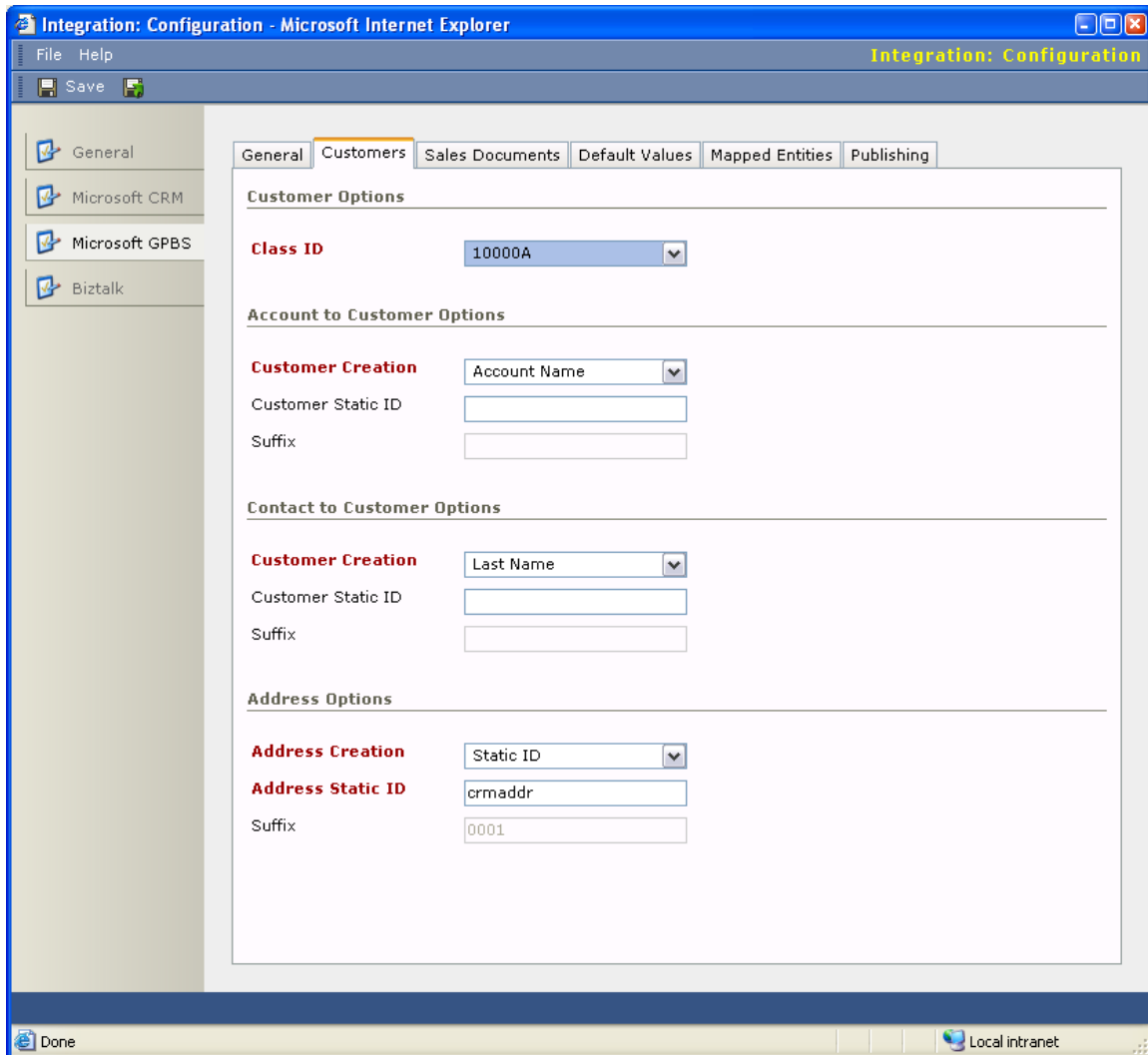

Figure 3: Screen shot from the Microsoft CRM Integration Settings and Administration tool.

The Settings and Administration tool divides into two main areas:

- Areas related to Microsoft CRM

- Areas related to Microsoft Business Solutions—Great Plains.

On the Microsoft CRM side tab, users will find all of the settings that affect data coming into Microsoft CRM. For example, there is a setting for whether contacts or accounts should be created when a customer has been created in Microsoft Business Solutions—Great Plains. Additionally, the business has the ability to set up what the default values should be for fields, such as shipping methods, payment methods, and other items that do not map directly to Microsoft Business Solutions—Great Plains.

If a user views the Microsoft Business Solutions—Great Plains side tab, the user will find all of the settings that affect data as it moves to Microsoft Business Solutions Financials applications. For example, the user can find a setting that determines what class ID should be used when creating new customers in Microsoft Business Solutions—Great Plains, as well as how the unique identifier of customers should be generated.

BizTalk document tracking

BizTalk Document Tracking is a stand-alone Web application that can be used to see the progress of documents that are being processed by Microsoft BizTalk Server. BizTalk Document Tracking allows a user to search by date or other criteria, or to build an advanced query using the Advanced Query Builder. The Advanced Query Builder enables a user to locate specific data within a document instance. For example, if a user had submitted Order Number FE-3425678 in Microsoft CRM and that number did not appear in Microsoft Business Solutions—Great Plains, then the user could query using the Advanced Query Builder.

Again, however, the real value of using Microsoft BizTalk Server lies in the fact that it offers an entire object model that the developer can use to create customer queries. For example, as will be detailed later, Microsoft CRM provides status on all integrated entities. If, however, a company wanted to provide more granular details to their end-users regarding integrated status, the company could do so using the object model provided by Microsoft BizTalk Server.

By default, Microsoft CRM Integration does not turn BizTalk Document Tracking on for any of the channels. To use document tracking, developers and administrators can turn on tracking on a per-channel basis. BizTalk Server does use database space for this feature, and there will also be a slight degradation in performance when channel tracking is turned on.

End-User Experience

State and Status of an integrated entity

The integration to external applications is known as "silent." That is, every time an order moves to another system (Microsoft CRM or Microsoft Business Solutions—Great Plains), the user is not notified that the order is being sent to another system. This silent integration is by design; the user does not need to be concerned that data is moving to other systems. However, Microsoft recognizes the need for a mechanism that will enable users to verify the status of those entities that have been integrated.

To accomplish this verification on all entities that can be integrated, Microsoft CRM provides two fields:

1. Back Office Integration Status
2. Back Office Integration Status Description.

While the implementation of the Status and Status Description fields is different from the status and status reason fields in other parts of Microsoft CRM, the purpose is identical. The Status field

details whether the integration has succeeded or failed, whereas the Status Description field details exactly what has occurred with the integrated entity.
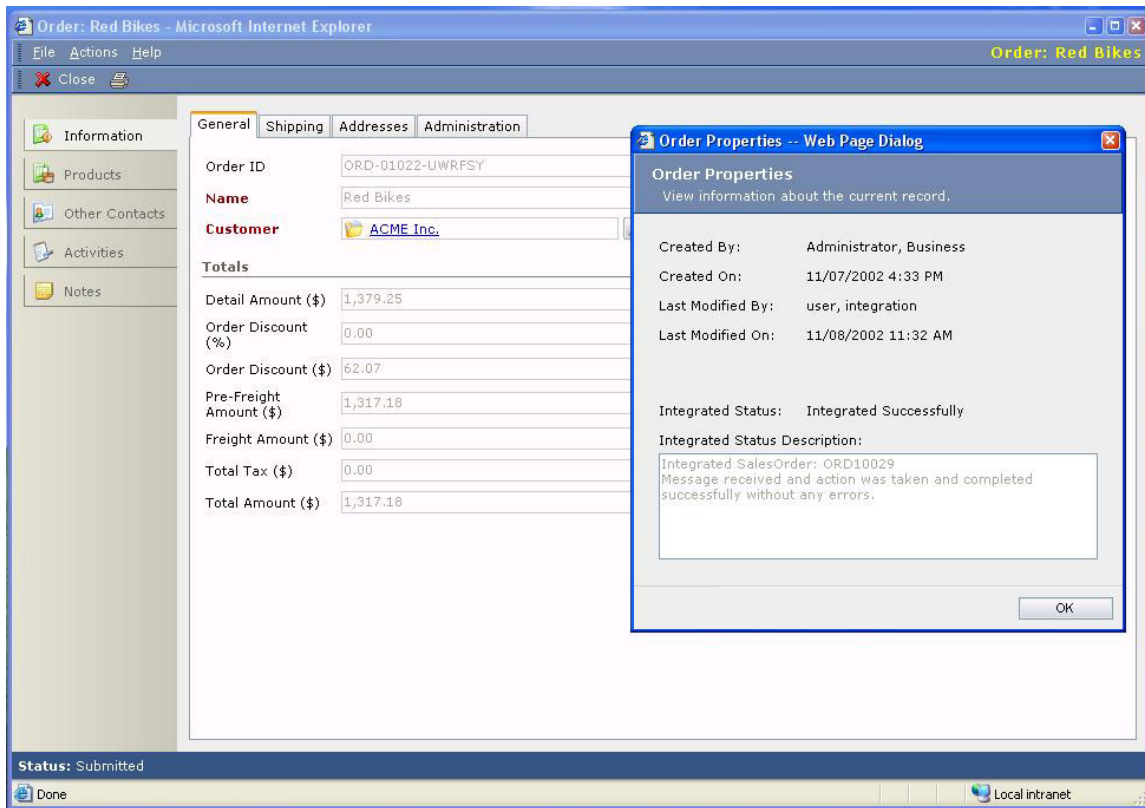

Figure 4: An account in Microsoft CRM with the Back Office Status and Status Description fields displayed.

Conclusion

Microsoft CRM Integration is built on Microsoft BizTalk Server 2002, enabling developers to create customized integrations with Microsoft Business Solutions—Great Plains Release 7.0. This initial release also includes components that can be used as the starting point for integration with third-party applications.

Future versions will offer enhanced integration between Microsoft CRM and Microsoft Business Solutions——Great Plains, as well as improvements to the platform that enables integration with third-party applications. Functionality and features planned for future versions include:

- Support for integrating with multiple "companies" within a single Microsoft Business Solutions—Great Plains deployment, as well as support for integrating orders in multiple currencies.

- An out-of-the-box integration with Microsoft Business Solutions—Solomon, Microsoft Business Solutions—Navision, and Microsoft Business Solutions—Axapta.

- Localized versions of Microsoft CRM Integration.

- Customization of events for which Microsoft CRM can publish data. For example, this would allow Microsoft CRM to publish leads out to a marketing automation system.

- Support for simultaneous integration with multiple systems. This would allow Microsoft CRM to be integrated, for example, with a financial application and a supply chain management application, and for the end user to be aware of the status in all integrated applications.

- Extensibility of the Microsoft CRM Integration Settings and Administration tool, so that ISV's will be able to offer a tool that acts as a control panel for the overall integration between Microsoft CRM and a 3rd party application.

Appendix A: Glossary of Technology Terms

Microsoft® Business Solutions Customer Relationship Management (Microsoft CRM) Integration makes use of Microsoft BizTalk® Server 2002 and other technologies, such as Microsoft Message Queuing (also known as MSMQ). This appendix provides a brief overview of technologies or components that are a part of BizTalk Server and other related technologies.

Application integration component (AIC): The piece of code that integrates BizTalk Server to a line-of-business application. An AIC is a Component Object Model (COM) component and can be created in any language that supports COM interface implementation, such as Microsoft Visual Basic® or Microsoft Visual C++® applications.

Port: A set of properties that determines which documents are transported to specific organizations or applications.

Channel: The processing agreement that ties together a configuration to completely specify the required processing for a particular message. A channel is dependent on a port to deliver the document to, a source document definition, and a destination document definition.

Message Queuing: A technology that enables applications running at different times to communicate across heterogeneous networks and systems that might be temporarily offline.

Receive function: A BizTalk Server component that can be configured to poll either a file folder or a queue. When the document arrives, the receive function uses the **IInterchange** object to submit the document to the messaging service layer of BizTalk Server.

VBScript: Allows for active scripting to a wide variety of environments, such as Internet Explorer, server scripting in Internet Information Services (IIS), and functoid scripting in BizTalk Server.

Pipeline Components: A pipeline component is a custom-tailored COM component that can extend the functionality of BizTalk Server 2002. Pipeline components are especially well suited for modeling linear processes, such as orders.

Appendix B: Overview of Integration Points

This appendix provides an overview of the integration points between Microsoft® Business Solutions Customer Relationship Management (Microsoft CRM) and Microsoft Business Solutions—Great Plains Release 7.0.

Integration Points

- Microsoft CRM Accounts and Contacts to Microsoft Business Solutions—Great Plains Customers
- Microsoft CRM Orders to Microsoft Business Solutions—Great Plains Orders (only create)

- Microsoft CRM Customer Addresses to Microsoft Business Solutions—Great Plains Customer Addresses
- Microsoft Business Solutions—Great Plains Customers to Microsoft CRM Accounts or Contacts
- Microsoft Business Solutions—Great Plains Customer Addresses to Microsoft CRM Customer Addresses
- Microsoft Business Solutions—Great Plains Inventory to Microsoft CRM Products
- Microsoft Business Solutions—Great Plains Units of Measure to Microsoft CRM Units of Measure
- Microsoft Business Solutions—Great Plains Price Lists to Microsoft CRM Discounts and Price Levels
- Microsoft Business Solutions—Great Plains Orders (SOP) to Microsoft CRM Orders
- Microsoft Business Solutions—Great Plains Invoices (SOP) to Microsoft CRM Invoices

Note

Products, pricing, discounts, and invoices that only move from Microsoft Business Solutions—Great Plains to Microsoft CRM are said to be mastered in Microsoft Business Solutions—Great Plains. This means that the user in Microsoft CRM cannot create, update, or delete products.