

Data Management

Microsoft Dynamics CRM 4.0

Sharing Data across Microsoft Dynamics CRM Deployments

Virtual Machine Build Guide

Date: March 2010



Acknowledgements

Initiated and sponsored by the Microsoft Dynamics CRM *Engineering for Enterprise* (MS CRM E²) Team, this document was developed with support from across the organization and in direct collaboration with the following:

Key Contributors

Brian Bakke (*Microsoft PFE*)
Grant Geiszler (*Microsoft PFE*)

Technical Reviewers

Roger Gilchrist (*Microsoft*)
Jim Steger (*Sonoma Partners*)

The MS CRM E² Team recognizes their efforts in helping to ensure delivery of an accurate and comprehensive technical resource in support of the broader CRM community.

MS CRM E² Contributors

Amir Jafri, Program Manager

Jim Toland, Content Project Manager

Feedback

Please send comments or suggestions about this document to the MS CRM E² Team feedback alias (entfeed@microsoft.com).

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

Legal Notice

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.

© 2010 Microsoft Corporation. All rights reserved.

Microsoft, Microsoft Dynamics, Microsoft Dynamics Logo, and SQL Server are trademarks of the Microsoft group of companies.

All other trademarks are property of their respective owners.

Table of Contents

- Overview 4
 - Transactional Replication 4
 - Data Sharing Strategy 4
 - Data Flow 5
 - Caveats 6
 - Supportability 6
- Virtual Hard Disk Requirements and Build-out Process 7
 - Baseline Microsoft Dynamics CRM 4.0 Environment 7
 - Pre-requisites for Setting up the Data Sharing Environment..... 7
 - Setting up the Data Sharing Environment 8
 - Phase 1: Configure Dynamics CRM for Multiple Tenants 8
 - Phase 2: Install and Configure Microsoft SQL Server Replication..... 10
 - Phase 3: Set Up Article Replication 15
 - Phase 4: Configure Database Triggers in the Subscriber Organization Databases 21
 - Phase 5: Conduct the Initial Snapshot 22
 - Phase 6: Configure Subscriber Organizations for the Replication Article 23
 - Phase 7: Customize Dynamics CRM..... 24
 - Phase 8: Perform Scenario Tests 26
- Appendix A: Changes Made to Entities 27
- Appendix B: Intel Hardware Components 29

Overview

This document explains how to configure the Microsoft Dynamics CRM 4.0 April 2009 VPC base image to enable data sharing between a Publisher organization and multiple Subscriber organizations. The configuration demonstrates how to use Microsoft Dynamics CRM to maintain referential integrity between multiple organizations by leveraging Microsoft SQL Server 2008 transactional replication.

Transactional Replication

Transactional replication typically starts with a snapshot of the publication database objects and data. After the initial snapshot is taken, subsequent data changes and schema modifications made at the Publisher are delivered to the Subscriber usually as they occur (in near real time). The data changes are applied to the Subscriber in the same order and within the same transaction boundaries as they occurred at the Publisher; therefore, within a publication, transactional consistency is guaranteed.

Transactional replication is typically used in server-to-server environments and is appropriate in each of the following cases:

- Incremental changes must be propagated to Subscribers as the changes occur.
- The application requires low latency between the time changes are made at the Publisher and the changes arrive at the Subscriber.
- The application requires access to intermediate data states. For example, if a row changes five times, transactional replication allows an application to respond to each change (such as firing a trigger), not simply the net data change to the row.
- The Publisher has a very high volume of insert, update, and delete activity.

By default, Subscribers to transactional publications should be treated as read-only, because changes are not propagated to the Publisher. However, transactional replication does offer options that allow for updates at the Subscriber.

Note: For more information about transactional replication, on MSDN, in SQL Server Books online, see *How Transactional Replication Works* at:

<http://msdn.microsoft.com/en-us/library/ms151706.aspx>

Data Sharing Strategy

The Publisher / Subscriber data sharing strategy is commonly used to maintain organizational consistency for certain entities and attributes of those entities. For this project, a single deployment was used with multiple organizations to show how a customer may deploy a Publisher / Subscriber organization relationship within a single deployment. The multiple Dynamics CRM organizations could also be deployed in separate deployments.

Consider the example of an organization that uses Microsoft Dynamics CRM as their primary sales force automation solution and that has provisioned multiple organizations for different teams within the organization. The Publisher organization will maintain important attributes of the Account entity such as industry SIC code and perhaps an attribute that is used to match the account record back to their ERP system. Within each team that is part of a Subscriber organization, however, different users may own the account record in each organization, depending on who is responsible for the territory in which that Account record resides. Additionally, the billing Contact for that account may be different in each Subscriber organization, the mailing address may be different, and the phone number may be different.

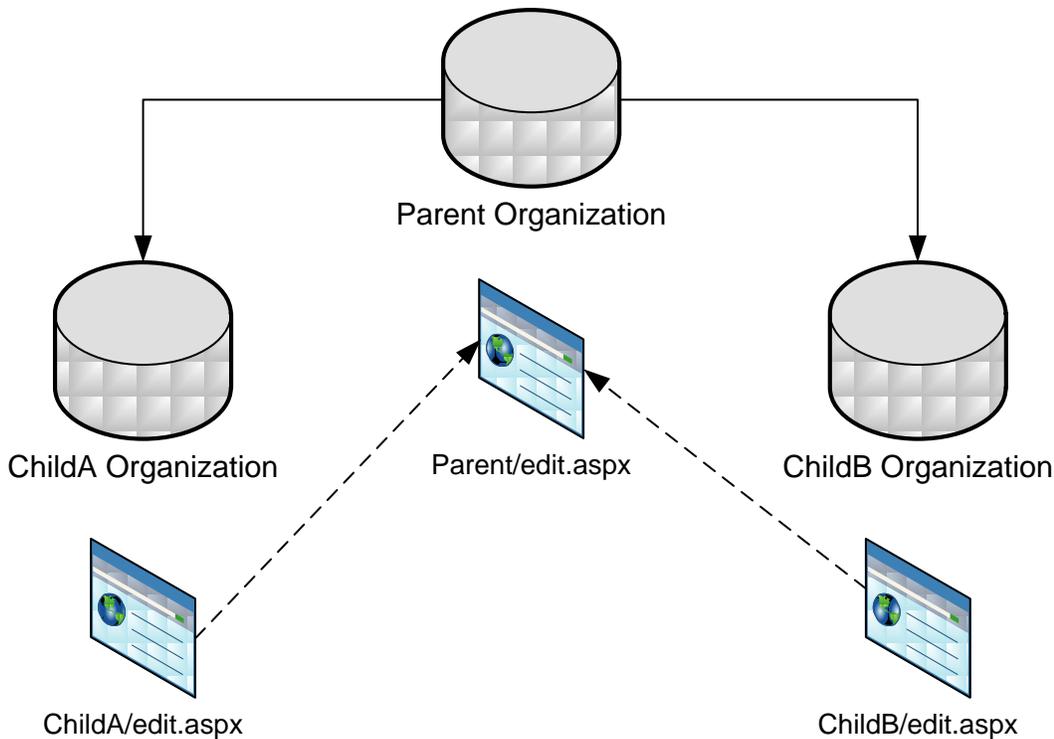
Data Flow

In this example, a few key attributes were identified on an entity to be read-only within Subscriber organizations. These attributes could only be modified within the Publisher organization and once a change was made to a record, those changes would be propagated from the Publisher organization to all Subscriber organizations.

Within Microsoft Dynamics CRM, in the Subscriber Organizations, these attributes were arranged in the same section on the form and their "read only" property was set to true. An "Edit" button was added to the Toolbar that would be used whenever a user needed to edit one of the read only attributes. When a user clicked this edit button, the main form from the Publisher organization would be displayed, where the attributes were available for modification.

After a row was modified at the Publisher, Microsoft SQL Server transactional replication would publish and distribute the change to all Subscriber organizations subscribing to that publication. The replication of the change would make the necessary modifications in the Subscriber organization only for the attributes that were identified to be read-only within the Subscriber organizations. To account for organization specific attributes such as **owninguser** and **organizationid**, triggers were implemented for all additions and modifications to provide the required values for those attributes.

The following diagram illustrates the data flow between organizations:



The steps in this document demonstrate techniques that can be used with Microsoft Dynamics CRM to accomplish these goals. These techniques provide a certain level of consistency across organizations in order to maintain "one version of the truth" while providing flexibility within Subscriber organizations to maintain attributes that are specific to their organization.

Important: The scenario described in this document assumes that creation of all new records occurs at the level of the Publisher organization.

Caveats

Be sure to consider the following caveats:

- Implementing this solution requires a Database Administrator and CRM Developer/Administrator with a thorough understanding of SQL Server Replication and the Dynamics CRM entity model. This is necessary to ensure that the correct set of data is replicated, all references are updated accordingly, security and authentication issues are considered, and the effect on CRM customizations is accounted for.
- Customization of CRM entities results in changes to the entity schemas, and replication must be suspended during this process. Note that the Publication may also need to be updated if the shared data schema has been modified.
- CRM benefits from holding data directly within its database
 - Advanced Find, Views and tools like the Reports Wizard can be used since all the data resides in Dynamics CRM
 - Workflows and Plug-ins can be triggered against data (as previously discussed in this paper)
 - Relationships between data can be established easily
- Managing Backup and Restore, and Disaster Recovery of this environment is also fairly complex. The procedures have to be managed in a way that in case of a failure, the data can be restored from a backup (at either the Publisher or Subscriber or both) and the replication relationship recreated and resynchronized.

Supportability

Commercially reasonable support will be provided for the SQL replication based solution described within this document. Commercially reasonable support is defined as all reasonable support efforts by Microsoft Customer Service and Support that do not require Microsoft Dynamics CRM code fixes. This support extends to Microsoft provided code and samples, but not to custom developed code which is the responsibility of the ISV or developer.

Virtual Hard Disk Requirements and Build-out Process

Baseline Microsoft Dynamics CRM 4.0 Environment

This project leverages as a baseline the environment provided in the Microsoft Dynamics CRM 4.0 April 2009 VPC release. Specifically, the virtual environment for this project relies upon the base virtual hard disk file **CRM-SRV-01 2009.vhd**.

Note: The Microsoft Dynamics CRM 4.0 April 2009 VPC is available from PartnerSource at: https://mbs.microsoft.com/Cms/Templates/document/General.aspx?NRMODE=Published&NRNODEGUID={B37438F6-44ED-4180-86C5-7FB218CFF223}&NRORIGINALURL=/partnersource/deployment/methodology/vpc/MSD_CRM4_VPCApril09.htm&NRCACHEHINT=Guest&wa=wsignin1.0

Pre-requisites for Setting up the Data Sharing Environment

Before proceeding with configuring the VPC for multi-organization data sharing, ensure that you have the components and information necessary to succeed by consider the following:

- Download and set up the Microsoft Dynamics CRM 4.0 April 2009 VPC, and verify that:
 - Microsoft Dynamics CRM 4.0 is installed and operational
 - Microsoft SQL Server 2008 is installed and operational
- Verify that the SQL Server Agent (MSSQLSERVER) service is running; if necessary, start the service before beginning the setup process.
- Review the basic concepts of SQL Server Replication – prior experience with configuring SQL Server Replication can be helpful.
- Review the SQL Server Replication Security requirements as outlined in the article *Replication Security Best Practices*, which is available from the MSDN SQL Server Developer Center at: <http://msdn.microsoft.com/en-us/library/ms151227.aspx>
- Determine which data needs to be replicated between organizations by defining the:
 - Publisher organization database and tables you wish to replicate
 - Subscriber organizations that you want to replicate data to
- Identify the tables that will be written to for Data Manipulation Language (DML) operations to entities that are configured for synchronization. To identify these tables, use SQL Server Profiler to trace the events that are generated from the CRM platform when creating, updating, or inserting data in CRM. After collecting the trace files, analyze them to determine which tables must be included in the replication article.

Note: For this project, the replication article included the following tables:

- | | |
|-----------------------|-------------------|
| ▪ AccountBase | ▪ LeadAddressBase |
| ▪ ContactBase | ▪ LeadBase |
| ▪ CustomerAddressBase | ▪ ProductBase |
- Review *Appendix A: Changes Made to Entities* to understand the configuration changes that are made to the virtual environment for this project.
 - Apply latest Dynamics CRM Update rollup to the environment.

Setting up the Data Sharing Environment

Phase 1: Configure Dynamics CRM for Multiple Tenants

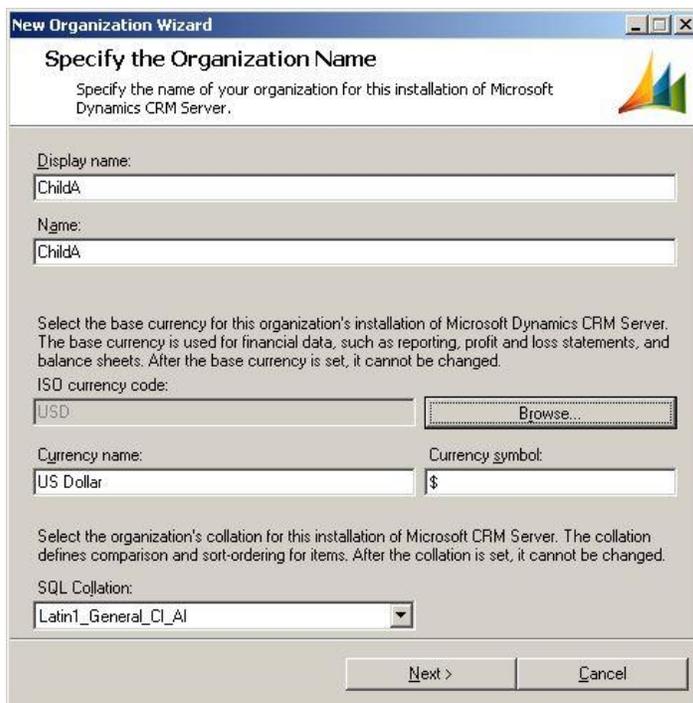
While the default base image includes a single tenant (organization), this project requires multiple organizations. For the purposes of this project, the default organization (Contoso) is the Publisher Organization and two Subscriber organizations (ChildA, ChildB) were created.

► To create an additional organization, perform the following steps:

1. Log on to the VPC image by using the following credentials:
 - User name: **administrator**
 - Password: **pass@word1**
 - Log on to: **CONTSO**
2. In the Dynamics CRM Deployment Manager, in the navigation pane, expand **Deployment Manager** if necessary, right-click **Organizations**, and then click **New Organization**.

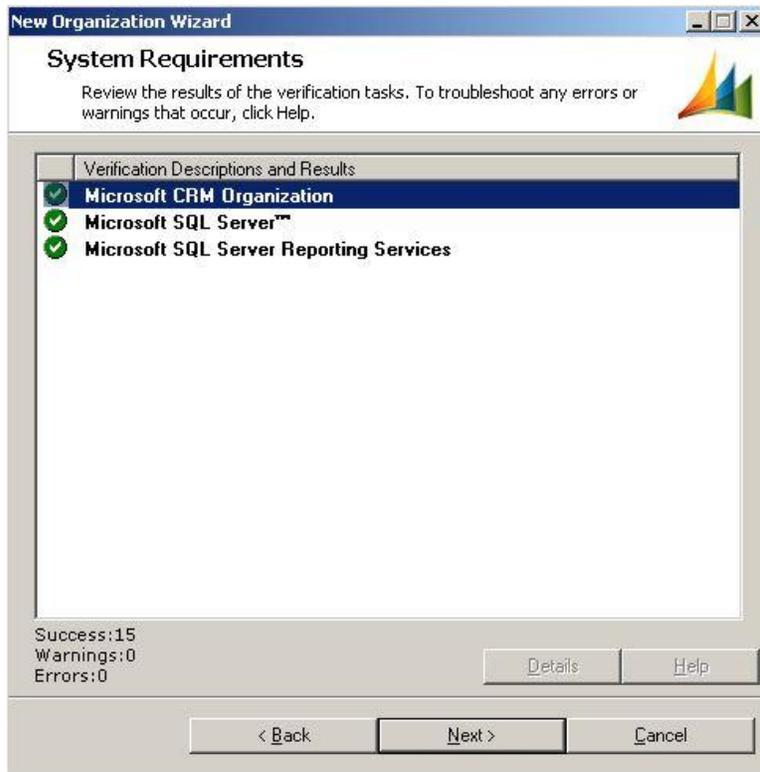


3. In the New Organization wizard, on the **Specify the Organization Name** page, specify the display name, name, and the currency information, and then click **Next**.



4. On the **Help Us Improve the Customer Experience** page, click **Next**.
5. On the **Select SQL Server** page, in the **SQL Server** drop-down box, select **CRM-SRV-01**, and then click **Next**.

6. On the **Specify Reporting Services Server** page, in the **Report Server URL** text box, type **http://CRM-SRV-01/ReportServer** and then click **Next**:
7. On the **System Requirements** page, view the results of the verification tasks, if necessary, troubleshoot any errors that occur, and then click **Next**.



8. On the **Ready to Install** page, verify the accuracy of the information displayed, and then click **Create**.
9. After the new organization is created successfully, on the **New Organization** page, click **Finish**.

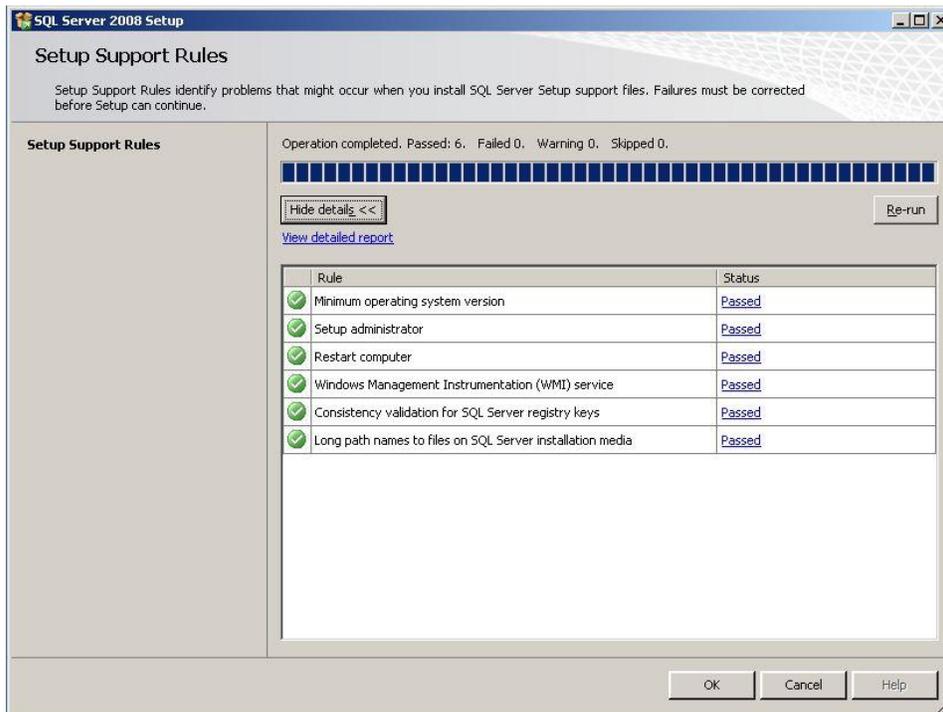


10. Repeat Steps 2 through 9 to create the second Subscriber organization (ChildB).
11. After successfully creating the two Subscriber organizations, close the Dynamics CRM Deployment Manager.

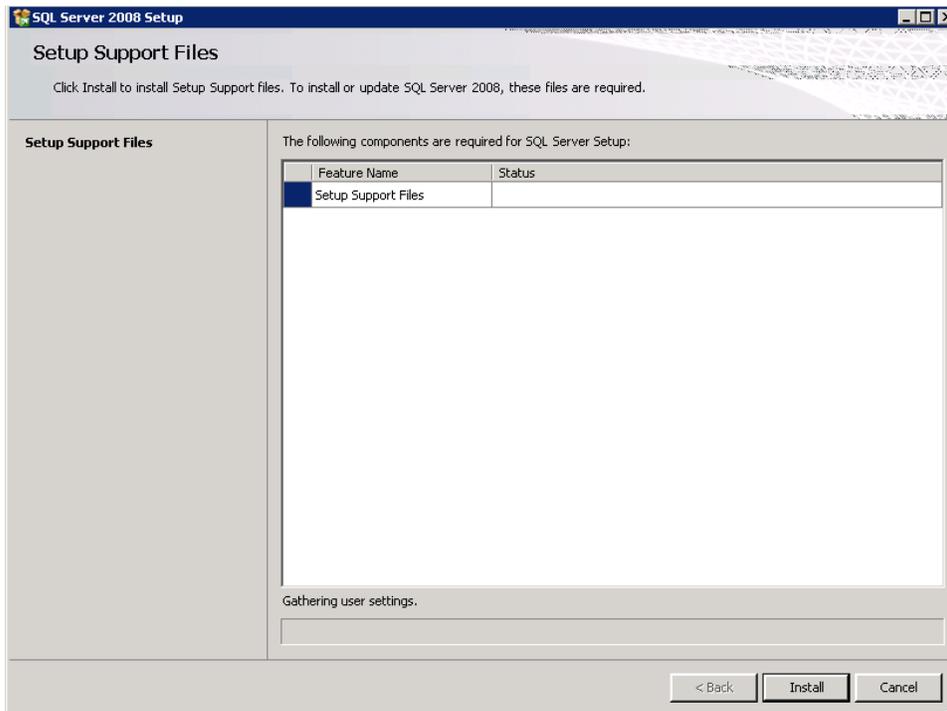
Phase 2: Install and Configure Microsoft SQL Server Replication

This project leveraged SQL Server Replication to replicate data from the Publisher to the Subscriber organizations.

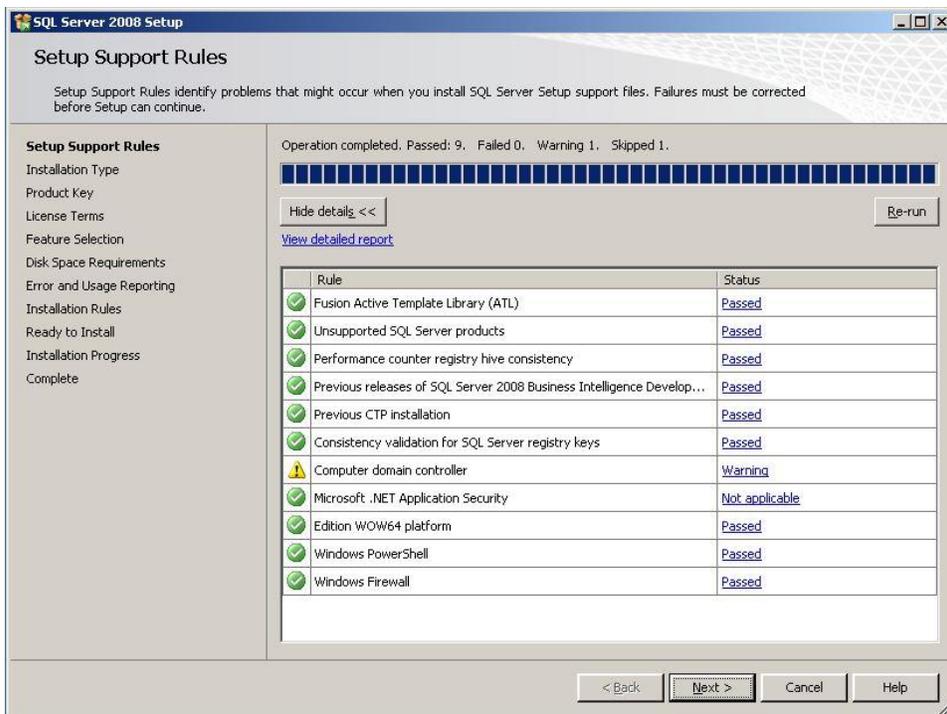
- ▶ To install SQL Server 2008 replication components on a computer running Microsoft SQL Server 2008, perform the following steps:
 1. In Control Panel, in the **Add or Remove Programs** window, under **Currently installed programs and updates**, click **Microsoft SQL Server 2008**, and then click **Change/Remove**.
 2. In the **SQL Server 2008** dialog box, click **Add**.
 3. In the **Browse For Folder** dialog box, navigate to the location of the Microsoft SQL Server 2008 installation files, and then click **OK**.
 4. In the SQL Server 2008 Setup wizard, on the **Setup Support Rules** page, click **Show details**, ensure that the Setup Support Rules operation indicates the passage of each rule, and then click **OK**.



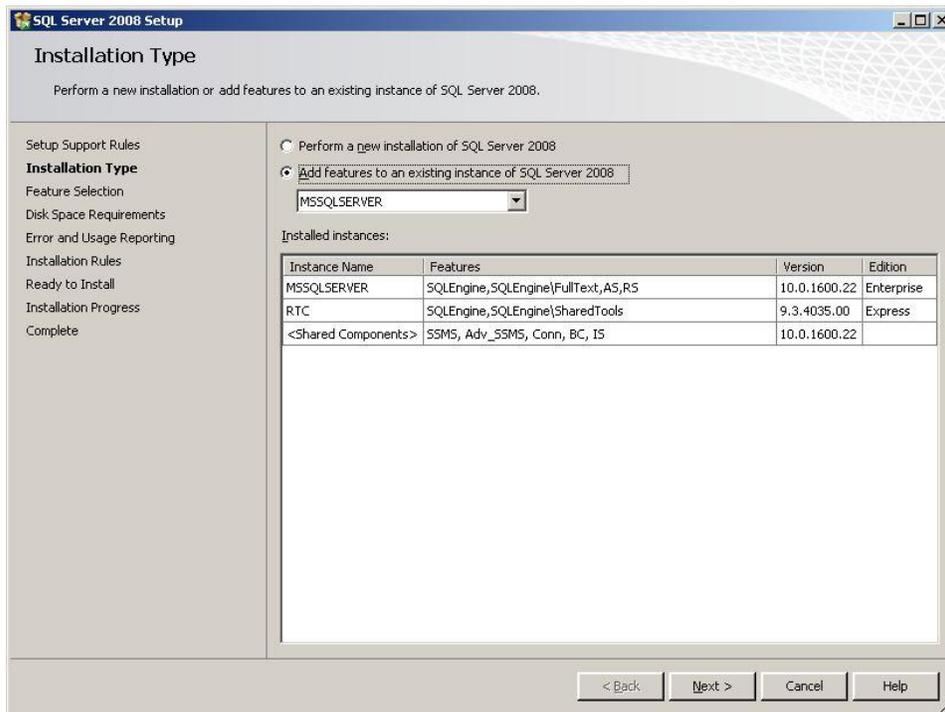
- To install the necessary SQL Server 2008 Setup Support files, on the **Setup Support Files** page, click **Install**.



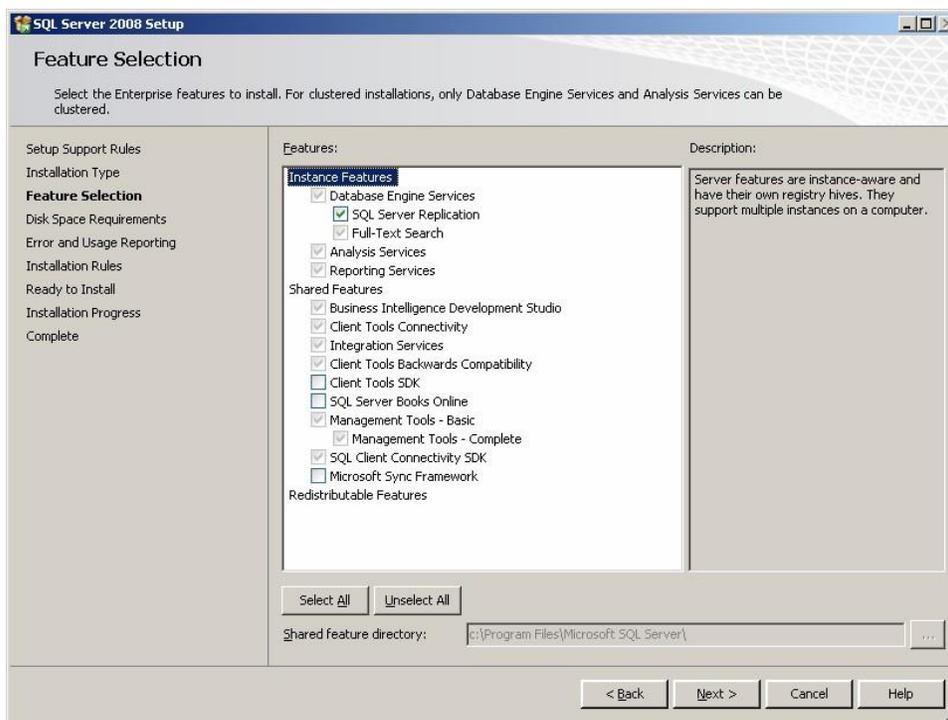
- On the **Setup Support Rules** page, ensure that the operation completed without any failures, and then click **Next**.



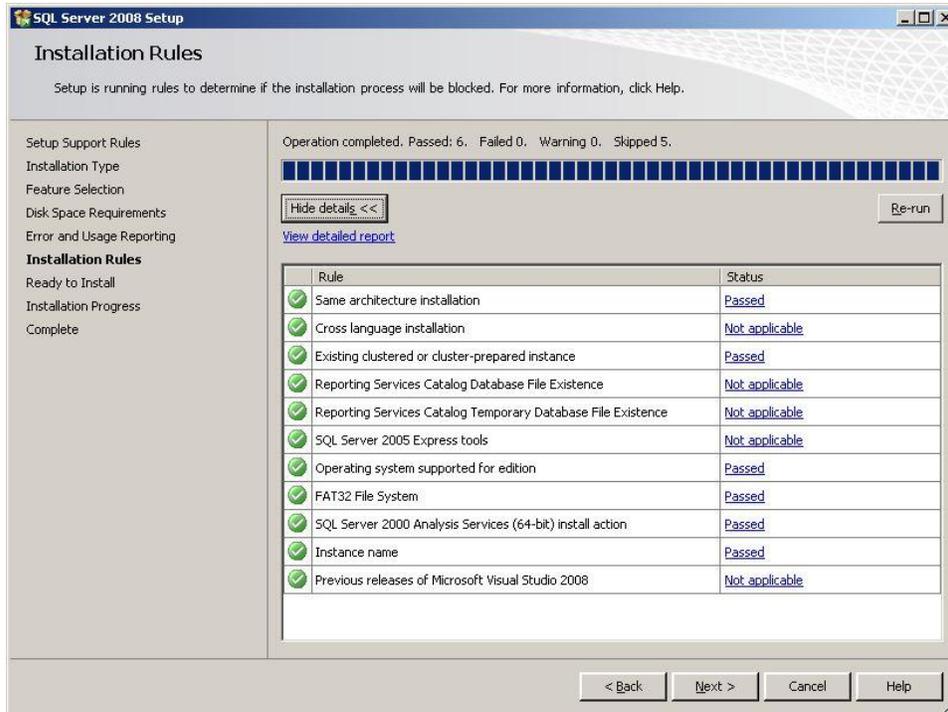
- On the **Installation Type** page, select the **Add features to an existing instance of SQL Server 2008** radio button, verify that **MSSQLSERVER** displays in the drop-down list box, and then click **Next**.



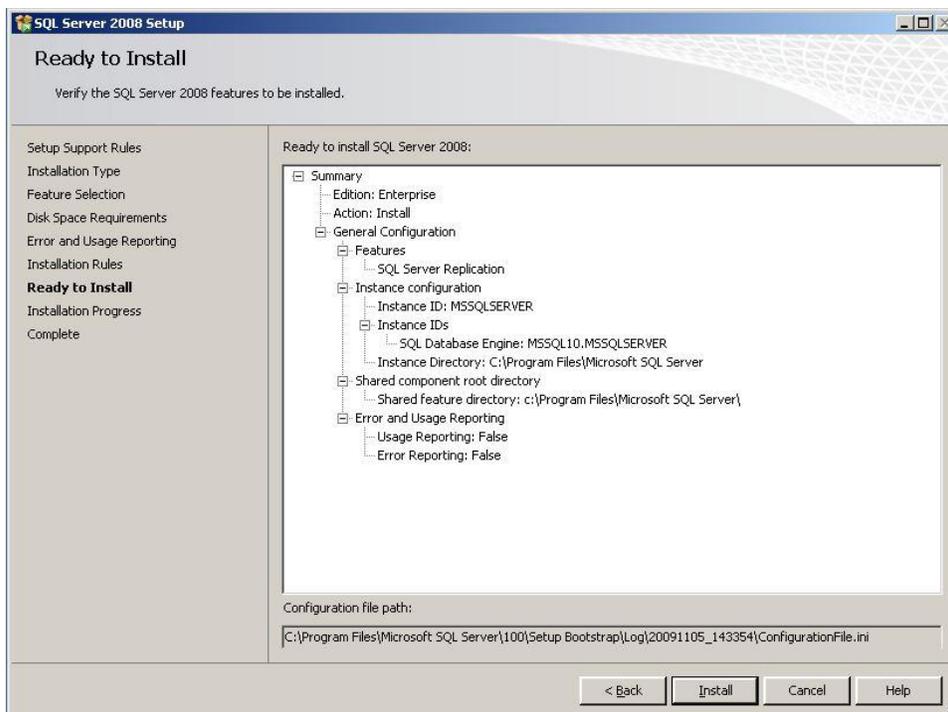
- On the **Feature Selection** page, under **Instance Features**, under **Database Engine Services**, select the **SQL Server Replication** check box, and then click **Next**.



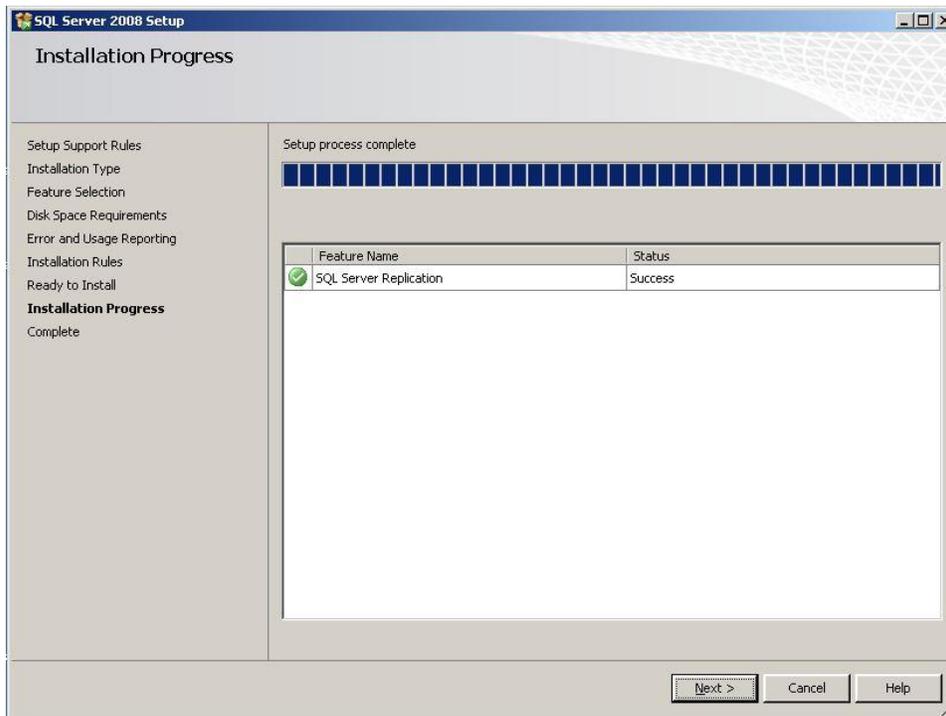
9. On the **Disk Space Requirements** page, review the disk usage summary, and then click **Next**.
10. On the **Error and Usage Reporting** page, click **Next**.
11. On the **Installation Rules** page, click **Show details**, ensure that the operation completed without any failures, and then click **Next**.



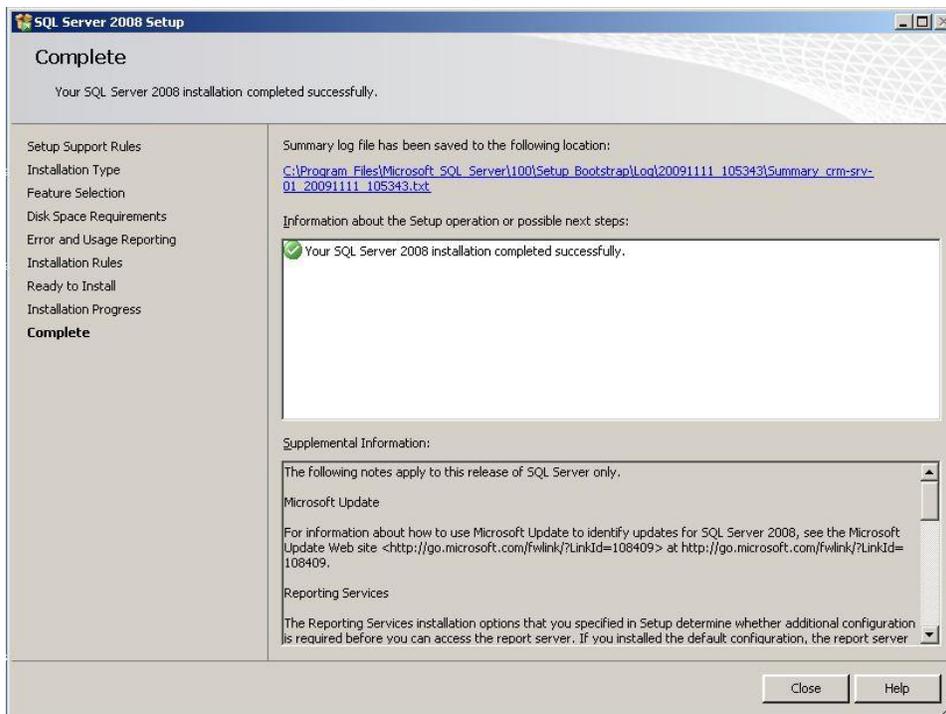
12. On the **Ready to Install** page, review and verify the features that will be installed, and then click **Install**.



13. On the **Installation Progress** page, ensure that the installation completed successfully, and then click **Next**.



14. On the **Complete** page, review the information about the successful installation, click **Close**, and then close **Add or Remove Programs**.



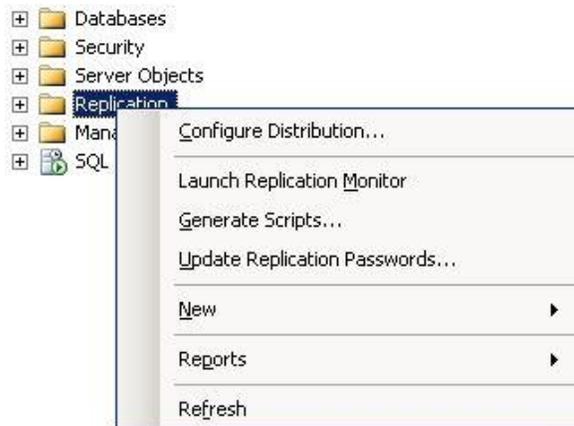
Phase 3: Set Up Article Replication

Proper function of the data replication process requires that an Article for replication be configured first. However, before a database owner can create a publication based on that database, an administrator must enable the database for publishing.

- ▶ To enable publishing for the Publisher database, perform the following steps:
 1. Launch Microsoft SQL Server 2008 Management Studio.
 2. In the **Connect to Server** dialog box, in the **Server name** text box, type **CRM-SRV-01** if necessary, and then click **Connect**.



3. In Microsoft SQL Server Management Studio, in the navigation pane, right-click **Replication**, and then click **Configure Distribution**.



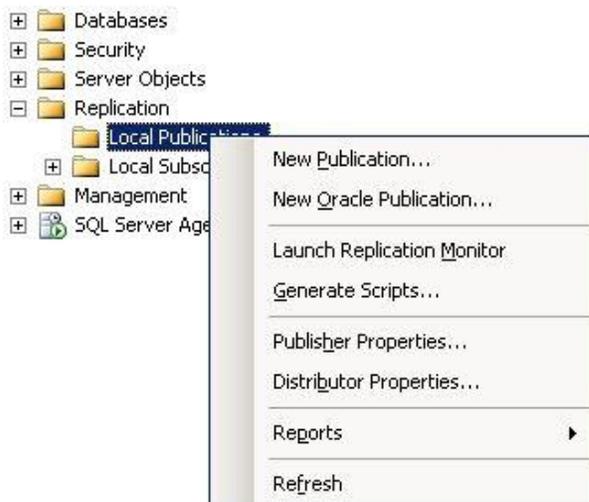
4. In the Configure Distribution wizard, click **Next** to display the **Distributor** page, verify that the **'CRM-SRV-01' will act as its own Distributor...** radio button is selected, and then click **Next**.
5. On the **SQL Server Agent Start** page, verify that the SQL Server Agent is configured to start automatically, and then click **Next**.
6. On the **Snapshot Folder, Distribution Database, Publishers, and Wizard Actions** pages, accept the default values by clicking **Next**.
7. On the **Complete the Wizard** page, review the options with which distribution will be configured, and then click **Finish**.
8. On the **Configuring...** page, verify that the operation completes successfully, and then click **Close**.

After the Publisher database is enabled for publishing, the database owner can configure the replication article.

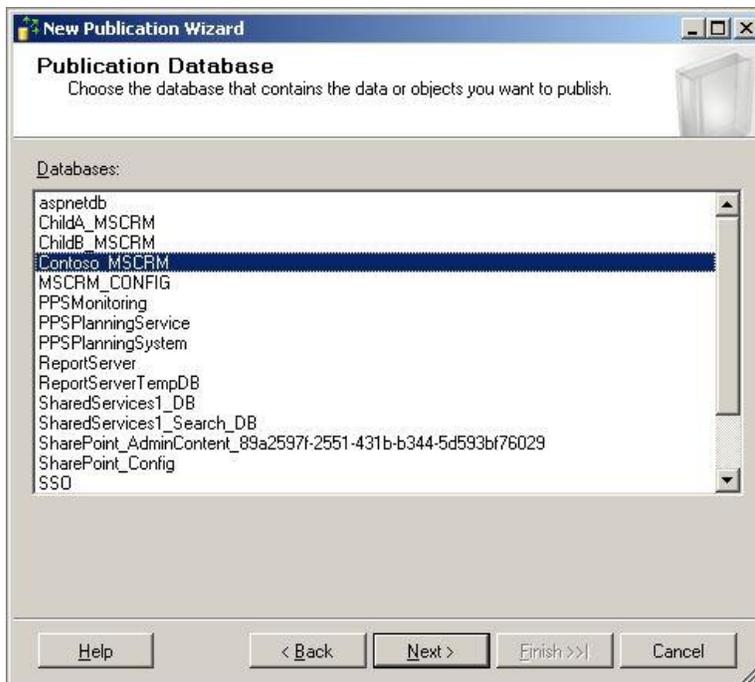
► To configure the replication article, perform the following steps:

Important: The following procedure illustrates setting up the “AccountBase” table as a replication article. To set up replication in your environment, you have the option of performing this procedure one time, including all the tables that you have previously identified for replication (as outlined in the pre-requisites), or multiple times, one time for each of the tables from which you want to replicate data.

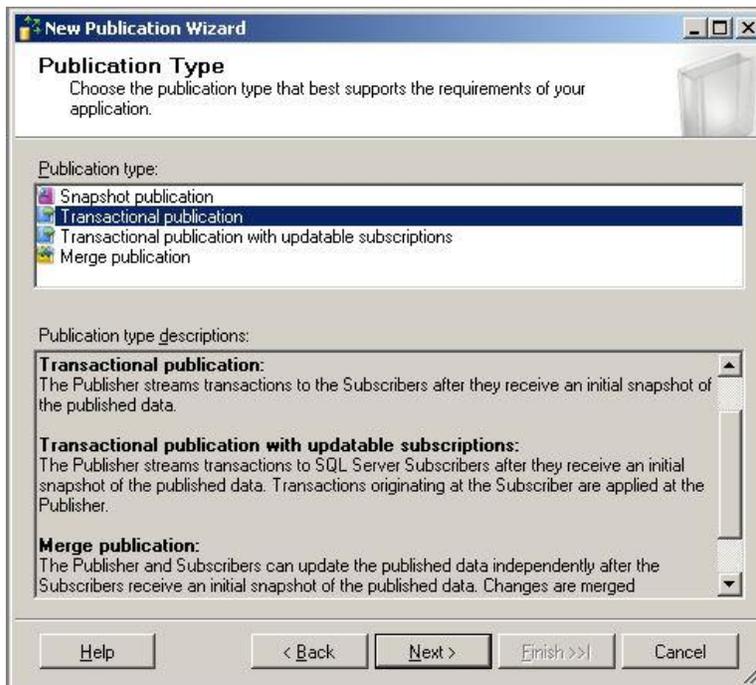
1. In SQL Server Management Studio, in the navigation pane, expand **Replication**, right-click **Local Publications**, and then click **New Publication**.



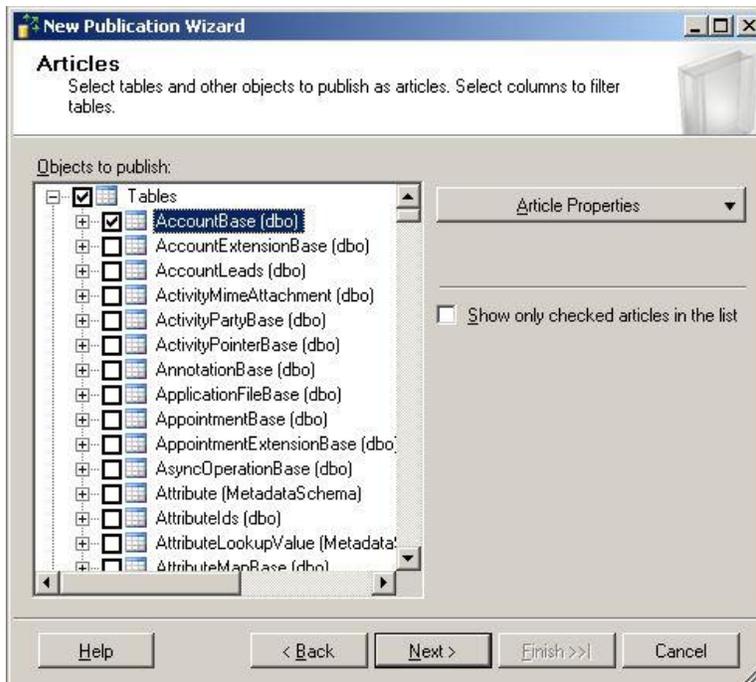
2. In the New Publication wizard, click **Next** to display the **Publication Database** page, under **Databases**, click **Contoso_MSCRM** (or the database you decide to use as your Publisher organization database), and then click **Next**.



3. On the **Publication Type** page, under **Publication type**, click **Transactional publication**, and then click **Next**.



4. On the **Articles** page, under **Objects to publish**, expand **Tables**, select the check boxes that are associated with the tables that you want to publish as articles (the "AccountBase" table in this example), and then click **Next**.

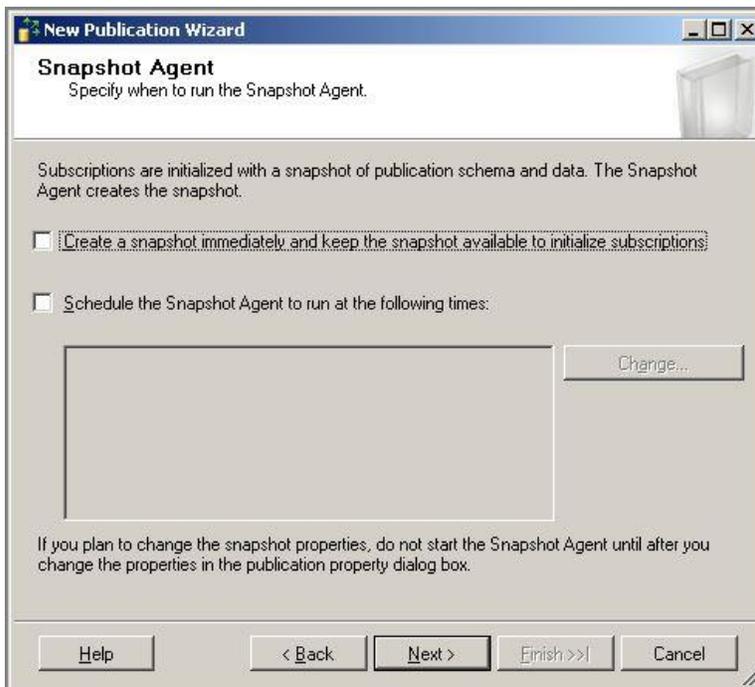


- On the **Filter Table Rows** page, verify that no tables are selected for filtering, and then click **Next**.



Important: It is recommended not to filter data in replicated tables unless required by the business rules associated with a specific scenario.

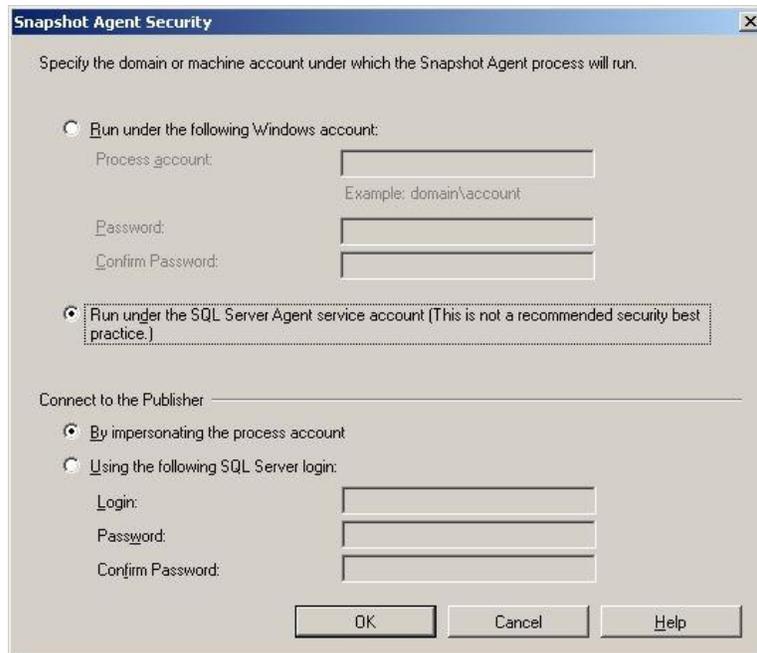
- On the **Snapshot Agent** page, verify that neither check box is selected (the associated database triggers have yet to be configured), and then click **Next**.



- On the **Agent Security** page, to configure the security settings for the snapshot and log reader agents, click **Security Settings**.
- In the **Snapshot Agent Security** dialog box, select the **Run under the SQL Server Agent service account...** radio button, and then click **OK**.

Important: While this project uses the SQL Server Agent Service Account, be sure to specify the domain or machine account that is most appropriate for your scenario. For additional information, on MSDN, in SQL Server Books Online, see the article *Replication Security Best Practices* at:

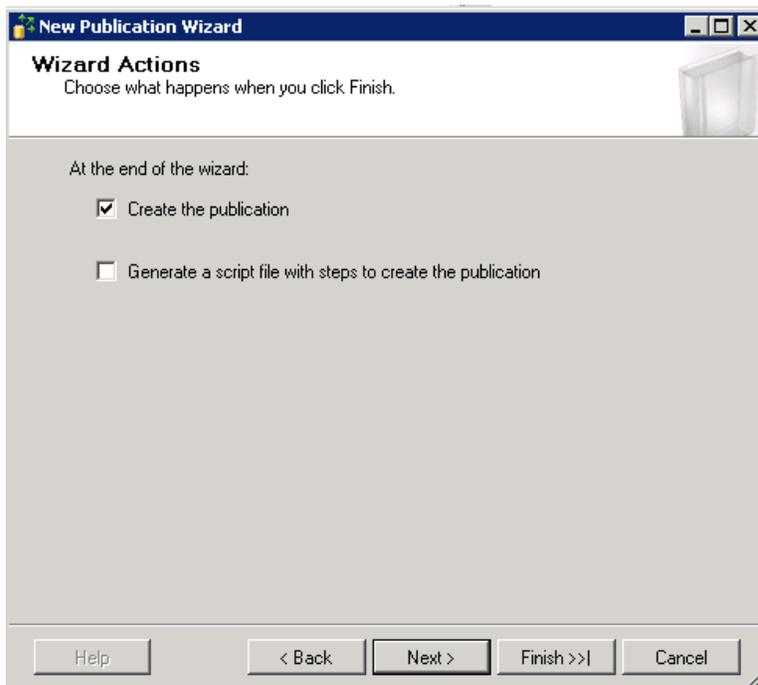
<http://msdn.microsoft.com/en-us/library/ms151227.aspx>



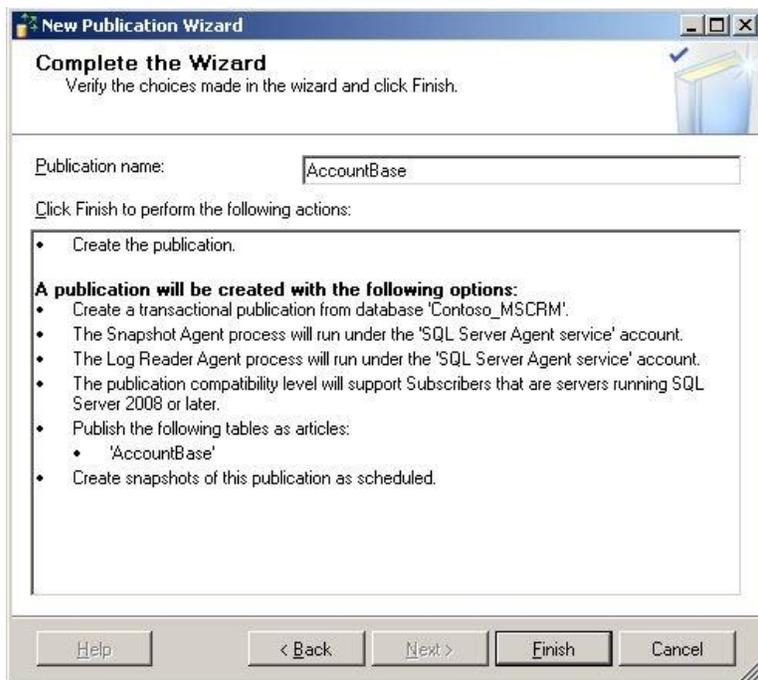
- On the **Agent Security** page, click **Next**.



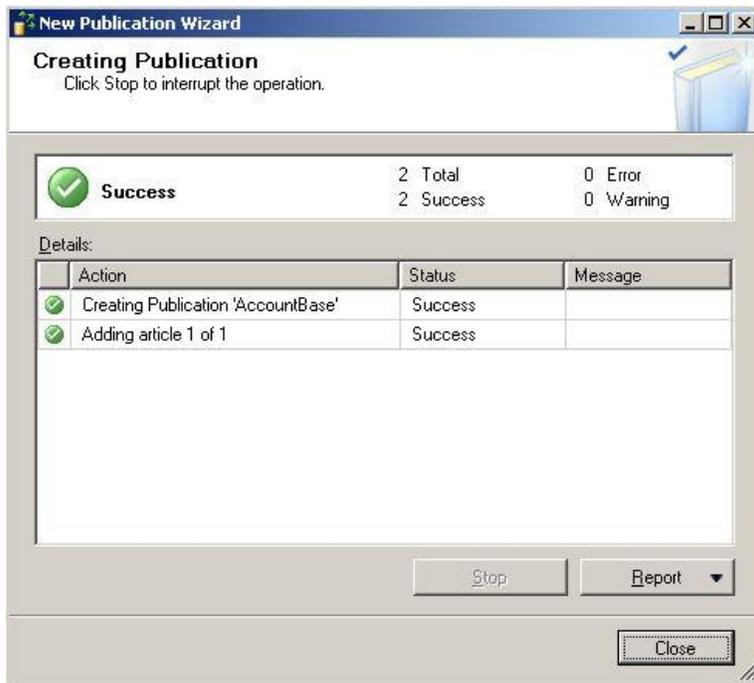
10. On the **Wizard Actions** page, verify that the **Create the publication** check box is selected, and then click **Next**.



11. On the **Complete the Wizard** page, in the **Publication name** text box, type **AccountBase** (the name of the table specified for replication), review the options you have selected for the publication, and then click **Finish**.



- On the **Creating Publication** page, verify that the operation completed successfully, and then click **Close**.



Phase 4: Configure Database Triggers in the Subscriber Organization Databases

When changes are made to a Subscriber organization by a Publisher organization, the **systemuserid** and **organizationid** attributes associated with those changes will be organization-specific. To account for the differences in these attributes across fields in Publisher and Subscriber databases, use INSTEAD OF database triggers to supply the correct values for those fields.

Important: The scenario described in this document assumes that creation of all new records occurs at the level of the Publisher organization.

The following example is a DDL statement for the proposed trigger:

```
create trigger setDefaultSecurityValuesForAccounts
on AccountBase
instead of insert
as
declare @adminUser uniqueidentifier
declare @bizUnitId uniqueidentifier

select @adminUser = SystemUserId from SystemUserBase where DomainName =
'CONTOSO\administrator'
select @bizUnitId = BusinessUnitId from SystemUserBase where DomainName =
'CONTOSO\administrator'

insert accountbase
(AccountCategoryCode, AccountClassificationCode, AccountId, AccountNumber, AccountRatingC
ode, Aging30, Aging30_Base, Aging60, Aging60_Base, Aging90, Aging90_Base, BusinessTypeCode,
CreatedBy, CreatedOn, CreditLimit, CreditLimit_Base, CreditOnHold, CustomerSizeCode, Custom
erTypeCode, DefaultPriceLevelId, DeletionStateCode, Description, DoNotBulkEMail,
```

```

DoNotBulkPostalMail, DoNotEMail, DoNotFax, DoNotPhone, DoNotPostalMail, DoNotSendMM, EMailAd
dress1, EMailAddress2, EMailAddress3, ExchangeRate, Fax, FtpSiteURL, ImportSequenceNumber,
IndustryCode, IsPrivate, LastUsedInCampaign, MarketCap, MarketCap_Base, MasterId, Merged, Mo
difiedBy, ModifiedOn, Name, NumberOfEmployees, OriginatingLeadId, OverriddenCreatedOn,
OwnershipCode, OwingBusinessUnit, OwingTeam, OwingUser, ParentAccountId, ParticipatesIn
Workflow, PaymentTermsCode, PreferredAppointmentDayCode, PreferredAppointmentTimeCode,
PreferredContactMethodCode, PreferredEquipmentId, PreferredServiceId, PreferredSystemUse
rId, PrimaryContactId, Revenue, Revenue_Base, SharesOutstanding, ShippingMethodCode,
SIC, StateCode, StatusCode, StockExchange, Telephone1, Telephone2, Telephone3, TerritoryCode
, TerritoryId, TickerSymbol, TimeZoneRuleVersionNumber, TransactionCurrencyId,
UTCConversionTimeZoneCode, WebSiteURL, YomiName)
select
AccountCategoryCode, AccountClassificationCode, AccountId, AccountNumber, AccountRatingCo
de, Aging30, Aging30_Base, Aging60, Aging60_Base, Aging90, Aging90_Base, BusinessTypeCode,
@adminUser, CreatedOn, CreditLimit, CreditLimit_Base, CreditOnHold, CustomerSizeCode, Custo
merTypeCode, DefaultPriceLevelId, DeletionStateCode, Description, DoNotBulkEMail,
DoNotBulkPostalMail, DoNotEMail, DoNotFax, DoNotPhone, DoNotPostalMail, DoNotSendMM, EMailAd
dress1, EMailAddress2, EMailAddress3, ExchangeRate, Fax, FtpSiteURL, ImportSequenceNumber,
IndustryCode, IsPrivate, LastUsedInCampaign, MarketCap, MarketCap_Base, MasterId, Merged, @a
dminUser, ModifiedOn, Name, NumberOfEmployees, OriginatingLeadId, OverriddenCreatedOn,
OwnershipCode, @bizUnitId, OwingTeam, @adminUser, ParentAccountId, ParticipatesInWorkflow
, PaymentTermsCode, PreferredAppointmentDayCode, PreferredAppointmentTimeCode,
PreferredContactMethodCode, PreferredEquipmentId, PreferredServiceId, PreferredSystemUse
rId, PrimaryContactId, Revenue, Revenue_Base, SharesOutstanding, ShippingMethodCode,
SIC, StateCode, StatusCode, StockExchange, Telephone1, Telephone2, Telephone3, TerritoryCode
, TerritoryId, TickerSymbol, TimeZoneRuleVersionNumber, TransactionCurrencyId,
UTCConversionTimeZoneCode, WebSiteURL, YomiName
from inserted

```

- ▶ To create the database triggers for the AccountBase table, perform the following steps:
 1. In SQL Server Management Studio, create a new query containing the database trigger text shown in the previous example.
 2. Run the query against the each of the ChildA_MSCRM and ChildB_MSCRM databases.

Phase 5: Conduct the Initial Snapshot

After creating the appropriate database triggers, conduct the initial snapshot to copy data from the Publisher to the Subscriber organizations by using any of various techniques, such as using SSIS packages, BCP, or the Management Studio Import Data feature.

To demonstrate how to copy data from the Publisher to the Subscribers will account for organization specific columns (**systemuserid** and **organizationid**) to avoid violating foreign key relationships, the following sample stored procedure is provided:

```

create procedure doInitialSnapshotAccount
as
insert ChildA_MSCRM..AccountBase (
AccountId, AccountCategoryCode, TerritoryId, DefaultPriceLevelId, CustomerSizeCode, Prefer
redContactMethodCode, CustomerTypeCode, AccountRatingCode, IndustryCode, TerritoryCode, Ac
countClassificationCode, DeletionStateCode, BusinessTypeCode, OwingBusinessUnit, OwingT
eam, OwingUser, OriginatingLeadId, PaymentTermsCode, ShippingMethodCode, PrimaryContactId

```

```
, ParticipatesInWorkflow, Name, AccountNumber, Revenue, NumberOfEmployees, Description, SIC,
OwnershipCode, MarketCap, SharesOutstanding, TickerSymbol, StockExchange, WebSiteURL, FtpSite
URL, EMailAddress1, EMailAddress2, EMailAddress3, DoNotPhone, DoNotFax, Telephone1, DoNotE
Mail, Telephone2, Fax, Telephone3, DoNotPostalMail, DoNotBulkEMail, DoNotBulkPostalMail, Cre
ditLimit, CreditOnHold, IsPrivate, CreatedOn, CreatedBy, ModifiedOn, ModifiedBy, ParentAccou
ntId, Aging30, StateCode, Aging60, StatusCode, Aging90, PreferredAppointmentDayCode, Preferr
edSystemUserId, PreferredAppointmentTimeCode, Merged, DoNotSendMM, MasterId, LastUsedInCam
paign, PreferredServiceId, PreferredEquipmentId, ExchangeRate, UTCConversionTimeZoneCode,
OverriddenCreatedOn, TimeZoneRuleVersionNumber, ImportSequenceNumber, TransactionCurrenc
yId, CreditLimit_Base, Aging30_Base, Revenue_Base, Aging90_Base, MarketCap_Base, Aging60_Ba
se, YomiName
```

```
)
```

```
select
```

```
a.AccountId, a.AccountCategoryCode, a.TerritoryId, a.DefaultPriceLevelId, a.CustomerSizeC
ode, a.PreferredContactMethodCode, a.CustomerTypeCode, a.AccountRatingCode, a.IndustryCod
e, a.TerritoryCode, a.AccountClassificationCode, a.DeletionStateCode, a.BusinessTypeCode,
a.OwningBusinessUnit, a.OwningTeam, a.OwningUser, a.OriginatingLeadId, a.PaymentTermsCode
, a.ShippingMethodCode, a.PrimaryContactId, a.ParticipatesInWorkflow, a.Name, a.AccountNum
ber, a.Revenue, a.NumberOfEmployees, a.Description, a.SIC, a.OwnershipCode, a.MarketCap, a.S
haresOutstanding, a.TickerSymbol, a.StockExchange, a.WebSiteURL, a.FtpSiteURL, a.EMailAddr
ess1, a.EMailAddress2, a.EMailAddress3, a.DoNotPhone, a.DoNotFax, a.Telephone1, a.DoNotEMai
l, a.Telephone2, a.Fax, a.Telephone3, a.DoNotPostalMail, a.DoNotBulkEMail, a.DoNotBulkPosta
lMail, a.CreditLimit, a.CreditOnHold, a.IsPrivate, a.CreatedOn, a.CreatedBy, a.ModifiedOn, a
.ModifiedBy, a.ParentAccountId, a.Aging30, a.StateCode, a.Aging60, a.StatusCode, a.Aging90,
a.PreferredAppointmentDayCode, a.PreferredSystemUserId, a.PreferredAppointmentTimeCode,
a.Merged, a.DoNotSendMM, a.MasterId, a.LastUsedInCampaign, a.PreferredServiceId, a.Preferr
edEquipmentId, a.ExchangeRate, a.UTCConversionTimeZoneCode, a.OverriddenCreatedOn, a.Time
ZoneRuleVersionNumber, a.ImportSequenceNumber, a.TransactionCurrencyId, a.CreditLimit_Ba
se, a.Aging30_Base, a.Revenue_Base, a.Aging90_Base, a.MarketCap_Base, a.Aging60_Base, a.Yom
iName
```

```
from Contoso_MSCRM..AccountBase a left join DefaultOrg_MSCRM..AccountBase b on
```

```
    a.AccountId = b.AccountId
```

```
where
```

```
    b.AccountId is null
```

► To conduct the initial snapshots, perform the following steps:

1. In SQL Server Management Studio, create a new query that contains the previous sample stored procedure.
2. Run the query against the each of the ChildA_MSCRM and ChildB_MSCRM databases.

Phase 6: Configure Subscriber Organizations for the Replication Article

After the initial snapshot has been completed and the triggers have been put in place, the organization can be configured as a subscriber to the previously created publication.

► To configure Subscriber organizations for the publication, perform the following steps:

1. In SQL Server Management Studio, right-click the publication to which you want to subscribe (**AccountBase** for this project), and then click **New Subscriptions**.
2. In the New Subscription wizard, if the welcome page displays, click **Next**.

3. On the **Publication** page, in the **Publisher** drop-down list, verify that **CRM-SRV-01** displays, under **Databases and publications**, expand **Contoso_MSCRM** and click on the publication to which you want to subscribe (**AccountBase** for this project) if necessary, and then click **Next**.
4. On the **Distribution Agent Location** page, for purposes of this project, verify that **Run all agents at the Distributor, CRM-SRV-01 (push subscriptions)** is selected, and then click **Next**.
5. On the **Subscribers** page, under **Subscribers and subscriptions databases**, under **Subscriber**, select the **CRM-SRV-01** check box, under **Subscription Database**, expand the drop-down list, select **ChildA_MSCRM**, and then click **Next**.
6. On the **Distribution Agent Security** page, configure the security accounts (for purposes of this project specify the SQL Server Agent service account) used to connect to the distributor and subscriber, and then click **Next**.
7. On the **Synchronization Schedule** page, under **Agent Schedule**, specify the appropriate synchronization schedule by using the drop-down list, and then click **Next**.
8. On the **Initialize Subscriptions** page, for the purposes of this project, deselect the **Initialize** check box, and then click **Next**.
9. On the **Wizard Actions** page, verify that the **Create the subscription(s)** checkbox is selected, and then click **Finish**.
10. On the **Complete the Wizard** page, verify the configuration options, and then click **Finish**.
11. On the **Creating Subscription(s)** page, verify that the subscription is created successfully, and then click **Close**.
12. For the purposes of this project, repeat the procedure above to configure the 'ChildB_MSCRM' database as a subscriber of the replication article, and then close SQL Server Management Studio.

With the Subscriber organizations configured as subscribers to the publication, use Microsoft Dynamics CRM to test the functionality to verify that the intended behavior is in production.

Phase 7: Customize Dynamics CRM

To ensure that the system maintains only a single instance of each record, the application forms in each organization can be updated to mark the shared fields as read-only. The application can also be customized to allow quick access to the associated record in the Publisher organization. To accomplish this, group all of the read-only fields into a separate section of the form for ease of use, and then create an Edit button on the form to provide an interface to the parent company.

- ▶ To add an 'Edit' button to the toolbar that is associated with the main form in each organization, perform the following steps:
 1. In Microsoft Dynamics CRM 4.0, navigate to **Settings>Customization>Export Customizations**.
 2. In the list of customizations, under **Display Name**, select **ISV Config**, click **Export Selected Customizations**, and then click **OK** to close the **Message from webage** warning.
 3. Save the exported file as **customizations.zip**.

4. Navigate to and extract the **customizations.zip** file, and then navigate to the extracted **customizations** folder.
5. Open the **customizations.xml** file with a text editor, and then within the account entity, after `<ToolBar ValidForCreate="0" ValidForUpdate="1">`, add text similar to the following (using the Publisher organization name in the URL):

```

        <Button Icon="/_imgs/ico_18_debug.gif"
        JavaScript="window.showModelessDialog('http://crm/Contoso/sfa/accts/edit.aspx?id='+crmForm.ObjectId, '', 'dialogHeight:564px;dialogWidth:1007px');">
            <Titles>
                <Title LCID="1033" Text="Edit" />
            </Titles>
            <ToolTips>
                <ToolTip LCID="1033" Text="Edit Read/Only Fields on this
Entity" />
            </ToolTips>
        </Button>

```

6. Save and close **customizations.xml**, and then navigate to **Settings>Customization>Import Customizations**.
 7. Navigate to and click **customizations.xml**, click **Upload**, click **Import Selected Customizations**, and then click **OK** to close the **Message from webpage** warning.
 8. In the Import Customizations... dialog box, verify that the customizations import successfully, and then click **OK**.
 9. Navigate to **Settings>Customization>Customize Entities**, under **Display Name**, click **Account**, and then click **Publish**.
- To add a 'Required Fields' section to the Account entity form, perform the following steps:
1. Navigate to **Settings>Customization>Customize Entities** if necessary, and then, under **Display Name**, double-click **Account**.
 2. In the **Entity: Account...** window, under **Details**, click **Forms and Views**, and then, under **Name**, double-click **Form**.
 3. Under **Common Tasks**, click **Add a Section**, in the **Name** text box, type **Read Only** and then click **OK**.
 4. Save the form.
- To make fields on the Account entity form read-only, perform the following steps:
1. If necessary, navigate to **Settings>Customization>Customize Entities**, and then, under **Display Name**, double-click **Account**.
 2. If necessary, in the **Entity: Account...** window, under **Details**, click **Forms and Views**, and then, under **Name**, double-click **Form**.
 3. For each field that you want to make read-only:
 - a. Select the field, and then, under **Common Tasks**, click **Change Properties**
 - b. In the **Field Properties** dialog box, under **Field Behavior**, select the **Field is read-only** checkbox.
 4. Under Location, in the **Section** drop down list, select **Read Only**, click **OK**, and then save and close the form.
 5. In the **Entity: Account...** window, under **Details**, click **Information** if necessary, in the **Actions** drop-down list, click **Publish**, and then save and close the entity.

Phase 8: Perform Scenario Tests

► Verify the solution by completing the following common test case scenarios:

1. Account Testing (From ChildA Organization)
 - a. Create an Account
 - b. Update an Account
 - c. Delete Account
 - d. Test replication with Subscriber offline
 - e. Test replication with Publisher offline
 - f. Repeat with ChildB Organization
2. Contact Testing (From ChildA Organization)
 - a. Create a Contact
 - b. Update a Contact
 - c. Delete a Contact
 - d. Test replication with Subscriber offline
 - e. Test replication with Publisher offline
 - f. Repeat with ChildB Organization
3. Product Testing (From ChildA Organization)
 - a. Create Product
 - b. Update Product
 - c. Delete Product
 - d. Test replication with Subscriber offline
 - e. Test replication with Publisher offline
 - f. Repeat with ChildB Organization

Appendix A: Changes Made to Entities

For this project, the changes made to entities are described in the following table:

Entity	Description of Changes
Account	<ul style="list-style-type: none"> ▪ Replicated the following columns: <ul style="list-style-type: none"> ○ AccountId ○ DeletionStateCode ○ Name ○ AccountNumber ○ VersionNumber ○ StateCode ▪ Changed the main application form by: <ul style="list-style-type: none"> ○ Making the Account Name and Account Number fields Read-Only ○ Organizing Read-Only fields into a section on the General tab ○ Adding "Edit" button that opens the same form on the Publisher ▪ Added database trigger to update the BusinessUnit and userId for each Subscriber organization ▪ Created stored procedure doInitialSnapshotAccounts
Contact	<ul style="list-style-type: none"> ▪ Replicated the following columns: <ul style="list-style-type: none"> ○ ContactId ○ DeletionStateCode ○ JobTitle ○ FirstName ○ LastName ○ VersionNumber ○ StateCode ▪ Changed the main application form by: <ul style="list-style-type: none"> ○ Making the First Name, Last Name, and Job Title fields Read-Only ○ Organizing Read-Only fields into their own section on the General tab ○ Adding "Edit" button that opens the same form on the Publisher ▪ Added database trigger to update the BusinessUnit and userId for each Subscriber organization ▪ Created stored procedure doInitialSnapshotContacts
CustomerAddressBase	<ul style="list-style-type: none"> ▪ Replicated the entire table; required for Accounts and Contacts

Entity	Description of Changes
Leads	<ul style="list-style-type: none"> ▪ Replicated entire table ▪ Changed the main application form by: <ul style="list-style-type: none"> ○ Making the Topic, First Name, Last Name, and Company Name fields Read-Only ○ Organizing Read-Only fields into their own section on the General tab ○ Adding "Edit" button that opens the same form on the Publisher ▪ Added database trigger to update the BusinessUnit and userId for each Subscriber organization ▪ Added a database trigger to replace the OwningUser and the businessUnit for inserts ▪ Created stored procedure doInitialSnapshotAccounts
LeadAddressBase	<ul style="list-style-type: none"> ▪ Replicated entire table; required for Leads
Product	<ul style="list-style-type: none"> ▪ Replicated the following columns: <ul style="list-style-type: none"> ○ ProductId ○ DefaultUoMScheduleId ○ DeletionStateCode ○ Name ○ DefaultUoMId ○ ProductTypeCode ○ Price ○ IsKit ○ ProductNumber ○ QuantityDecimal ○ IsStockItem ○ CreatedOn ○ ModifiedOn ○ StatusCode ○ VersionNumber ○ TransactionCurrencyId ○ StateCode ○ Price_Base ▪ Inserted data from the following tables <ul style="list-style-type: none"> ○ TransactionCurrencyBase ○ UoMBase ○ UoMScheduleBase ▪ Changed the main application form by: <ul style="list-style-type: none"> ○ Making the Name, ID, Default Unit, Decimals Supported, Unit Group, and List Price fields Read-Only ○ Organizing Read-Only fields into their own section on the General tab ○ Adding "Edit" button that opens the same form on the Publisher ○ Added database trigger to update the BusinessUnit and userId for each Subscriber organization

Appendix B: Intel Hardware Components

For the purposes of this project, the Premier Field Engineering team in Fargo had access to a variety of recent technology offerings from Intel Corporation. These technologies are described in greater detail in the following sections.

Important: None of the hardware or technologies described in this appendix is required to implement the solutions described in this document.

Intel® Xeon® 5500 Series Processors and the Nehalem Microarchitecture

The Intel® Xeon® Processor 5500 series brings together a number of innovative technologies to deliver intelligent performance:

- Intel® Turbo Boost Technology, together with Intel® Intelligent Power Technology, delivers performance on demand, letting processors operate above the rated frequency to speed specific workloads and reduce power consumption during low utilization periods.
- Intel® Hyper-Threading Technology+ benefits from larger caches and massive memory bandwidth, delivering greater throughput and responsiveness for multi-threaded applications.
- Intel® QuickPath Technology and an integrated memory controller speed traffic between processors and I/O controllers for bandwidth intensive applications, delivering up to 3.5x the bandwidth for technical computing.

Note: For additional information about these innovative technologies, please see:

<http://www.intel.com/Assets/PDF/prodbrief/xeon-5500.pdf>

The key benefits, technologies, and usage associated with the Intel® Xeon® Processor 5500 series are provided in the following table.

Area	Detail
Key Benefits	<ul style="list-style-type: none"> ▪ Up to 2.25x more performance for enterprise applications* ▪ Up to 50 percent lower system idle power²⁰ ▪ Up to 18 slots DIMM with up to 144 GB DDR3 memory ▪ Up to 42 lanes PCI Express*(36 lanes PCI Express* 2.0)
Key Technologies	<ul style="list-style-type: none"> ▪ Two Intel® Xeon® Processor 5500 series ▪ Intel® Turbo Boost Technology ▪ Intel® Hyper-Threading Technology ▪ 8 MB shared L3 cache featuring Enhanced Smart Cache ▪ Intel® QuickPath Technology ▪ Intel® Intelligent Power Technology ▪ Intel® Virtualization Technology
Key Usage	<ul style="list-style-type: none"> ▪ Exceptional performance and efficiency for general-purpose business computing including: <ul style="list-style-type: none"> ○ E-mail servers ○ Web servers ○ File server ○ Business applications ▪ Flexible infrastructure for virtualization

Intel® X25-E Extreme SATA Solid State Drives

The Intel X25-E Extreme SATA Solid-State Drive (SSD) features the latest-generation native SATA interface with an architecture using 10 parallel NAND flash channels equipped with single-level cell NAND flash memory lithography (50nm). With powerful Native Command Queuing to enable up to 32 concurrent operations, these Intel SSDs drastically outperform traditional hard disk drives.

SSDs contain no moving parts; they are silicon-based storage devices that are comprised of an array of NAND FLASH parts. Intel's SSD control logic can exploit parallelism, such that several small IO requests can be handled simultaneously, or one large request can be aggregated, across multiple NAND components.

From a solutions perspective, SSD-based storage provides significantly better price/performance and TCO benefits than do traditional hard disk drives (HDDs), especially for the right classes of application. For example, when handling random IO request traffic, HDDs have to seek out a new location for each operation, which can impose substantial latency.

The highly concurrent but typically unrelated queries found in many CRM applications (such as Microsoft Dynamics CRM 4.0) result in many random IO requests to the storage subsystem. Intel SSDs can provide very low response latency for these requests, even when multiple outstanding requests are queued at each device. This enables the SQL middleware running under the CRM application to operate more efficiently, allowing higher application throughput with less CPU resource utilization. In addition, because SSDs can handle many more IO requests than HDDs, the storage subsystem can be comprised of considerably fewer devices, reducing both space and thermal demands.