[downloaded from: http://support.microsoft.com/?id=282977]

Microsoft Jet 4.0

Frequently asked questions about how to use Microsoft Jet 4.0 replication with Microsoft Access 2000, Access 2002, and Access 2003 replication

Microsoft Product Support Services White Paper

Written by Michael Kaplan, Mary Chipman, Paul Litwin, Steve Thompson, and John Blaine

Published on February 17, 2006

Abstract

This white paper contains a list of frequently asked questions and answers about Microsoft Jet 4.0 Replication when you are using Access 2000, 2002, or 2003.

Template updated July 21, 2004

This is a preliminary document and may be changed substantially prior to final commercial release of the software described herein.

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, AS TO THE INFORMATION IN THIS DOCUMENT.

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

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

CONTENTS

INTRODUCTION

This white paper contains a list of frequently asked questions and answers about Microsoft Jet 4.0 Replication when you are using Access 2000, 2002, or 2003.

1. What is a GUID?

Replication must have a method to uniquely identify various items such as each record, table, and replica in a replicated database. The identifier is a globally unique identifier (GUID) that is created by the Microsoft Jet Database Engine to guarantee uniqueness. GUIDs are not unique to Jet or to replication. Many programs create and use GUIDs. For example, the Microsoft® Windows® operating system assigns a GUID to every ActiveX® control that is installed on your computer.

When a database is converted to a Design Master, several fields are added to each table. This includes a field that is named **s_GUID**. GUIDs are created by using a combination of the network node ID, a time value, a clock sequence value, and a version value. The following is an example of a GUID:

> 36B295B1-D128-11D0-81F9-0000F649884F

For more information about how GUIDs are created, see the *Database Replication in Microsoft Jet 4.0* white paper. To download this white paper, see the following Microsoft Knowledge Base article:

> Jet 4.0 Replication white papers available in MSDN Online Library
>
> http://support.microsoft.com/kb/190766

2. Can I use a GUID as a primary key field?

Yes, you can. However, it is not required that you do this, and we do not recommend it. If you want to use the Microsoft Access Conflict Viewer to resolve synchronization conflicts, you should not use a GUID as a primary key field. If you decide to use a GUID as the primary key for a table, you must do this before you create a replica set. To use an **AutoNumber** field as a primary key field in a replicated database, follow these steps:

1. Select the **AutoNumber** data type for the primary key field.
2. Select **ReplicationID** as the **FieldSize** property setting.

An additional problem that can occur when you use GUIDs for your primary key fields is that you may have problems trying to use GUIDs in DLOOKUP expressions, query parameters, subform master/child links, and other situations. It is best to avoid this practice.

3. I have decided to use a GUID field as the primary key for one of the tables in my replicated database. The Conflict Viewer will not open when that table contains conflicts. However, the Conflict Viewer works fine on tables that contain a random AutoNumber. Why?

When the database is converted to a replica, the Microsoft Jet Database Engine will use that **AutoNumber** field as the GUID and will not add the new **s_GUID** field. This is described in What is a GUID? If you decide to use a GUID as a primary key field for a table, and then replicate that table, the Conflict Viewer in Access cannot be used to resolve synchronization conflicts in that table. This behavior occurs because the Conflict Viewer relies on the **s_GUID** field. You will have to create a custom conflict resolving routine for that table. The Conflict Viewer will work on tables in the same database that do contain the **s_GUID** field.

As stated in Can I use a GUID as a primary key field?, it is best to avoid using GUID fields for primary keys.

4. Why did replication change all my AutoNumber fields to Random Increment?

By default, **AutoNumber** fields increment by 1 every time that a record is added. This does not work in a replicated application. If it did, each member of the replica set would produce new records independently of one another, and each replica set would contain identical primary key values. This causes duplicate primary key conflicts on every synchronization. Therefore, when you convert your regular database into a replica, this behavior is automatically changed to a randomly incrementing value between –2,000,000,000 and 2,000,000,000. This greatly reduces the chance of two replicas assigning the same primary key value. If you are using **AutoNumber** both as a primary key value and to number records sequentially, where the numbers are important in a business context such as an invoice number, you must consider other alternatives before converting your database into a replica.

Another method is to create a custom AutoNumber routine that assigns numbers by using code that is executed from your forms. This code either incorporates a site or replica ID as part of the primary key, or perhaps uses different sets of values for each replica. The following Microsoft Knowledge Base articles may help you implement a custom AutoNumber routine:

> How to create a multiuser custom counter
>
> http://support.microsoft.com/kb/210194
>
> How to increment the numeric portion of a string
>
> http://support.microsoft.com/kb/209830

5. Can I use a field that has a GUID in a criteria for a DLookup? In a parameter query? In a search?

You can use a GUID as part of the criteria for a **DLookup** function or for any one of the other domain functions in Access. You can use it in a parameter query in Access. You can also use it in a search.

The Jet Database Engine stores GUIDs as arrays of type Byte. However, Microsoft Access cannot return Byte data from a control on a form or from a report. To return the value of a GUID from a control, use one of the following methods:

- Convert it to a string. To convert a GUID to a string, use the **StringFromGUID** function. To convert a string to a GUID, use the **GUIDFromString** function. For more information about these functions, see the Access Help file.
- Use the **Text** property of the control instead of the **Value** property. When a GUID is bound to a control, the canonical (string) form of the GUID is displayed. The **Value** property is the binary data, and the **Text** property is the canonical form. Note that the control must have the focus to retrieve the GUID value by referring to the **Text** property of the control.

6. What are some design issues that I should consider with replication?

A book could probably be written on this subject alone!

Microsoft Jet enforces a 4096-byte limit and a 255-field limit to any record. These limits include fields and data that you add to tables yourself, and include any fields and data that Jet adds during replication. When you design tables in an application that you plan to replicate, make sure that you allow for the fields that Jet is likely to add.

At a minimum, every record in a replicated table receives an **s_GUID** field, an **s_Lineage** field, an **s_ColLineage** field for tables where column-level tracking has been set, and an **s_Generation** field. These fields collectively require approximately 32 bytes per record. This means that you should design tables that have no more than 251 fields (255 maximum, minus 4) and 4,064 bytes (4,096 maximum, minus 32). This is not generally a hindrance to well-designed applications, because very few well-designed applications use either the maximum allowed fields or bytes in a single record. If your application uses Memo or OLE Object fields (BLOB fields), replication will add an additional 4-byte field per BLOB field. Through an exchange optimization, only the long binary fields that have been modified are sent in an exchange.

Microsoft Jet replication does not support check constraints. Check Constraints is a new feature that is supported by Microsoft Jet 4.0. With this feature, you can have more complex constraints on and across tables.

The topology of your application should be designed around the real-time load and the estimated size of your database. For example, you should have more than one hub database if many replicas are synchronizing at the same time and many data changes are being exchanged during the synchronization. It is a good idea to prototype your application by using real-time data and load levels.

7. Why are some of my objects renamed?

If a table is created in Microsoft Access as the Design Master and it has the same name as a local object in any one of the replica members, the local object in the replica is renamed to "<*local object*>_Local" instead of triggering a design error.

8. How can I synchronize in a one-way direction?

Use Microsoft Jet OLE DB Provider and Replication Objects (JRO) and the **Synchronize** method. The default direction is bidirectional, but you can choose to synchronize in a one-way direction. When you choose to synchronize in a one-way direction, you can choose to only export changes or only import changes during synchronization instead of both. Consider that any design changes to Jet objects such as tables, queries, or relationships will be bidirectional.

The syntax is as follows:

```
Replica.Synchronize(Target [, SyncType] [, SyncMode])
```

The **Synchronize** method syntax has the following parts.

Part	**Description**	
Replica	An object variable that points to a replica object that you want synchronized.	
Target	A String value specifying the path and file name of the replica with which to synchronize, the name of the Synchronizer that manages the target replica, or the name of the Internet server that contains the target replica.	
SynchType	Optional. An **Enum** value specifying the type of synchronization to perform. The following values are valid for SyncType:	
	jrSyncTypeExport	Sends changes from the current replica to the target replica.
	jrSyncTypeImport	Sends changes from the target replica to the current replica.

	jrSyncTypeImpExp	Default. Sends changes from the current replica to the target replica and vice-versa.
SynchMode	Optional. An **Enum** value specifying the method of synchronization. The following values are valid for SyncMode:	
	jrSyncModeIndirect	Default. Indirect synchronization.
	jrSyncModeDirect	Direct synchronization.
	jrSyncModeInternet	Indirect synchronization over the Internet.

The following is a sample procedure to customize for your own use. The procedure expects the following:

- Database/path name to synchronize
- Path/database target name
- Synchronization type
- Synchronization mode

For the *intSynchType* argument you must specify **1** for export changes, **2** for import changes, or **3** for bidirectional changes. For the *intSynchMode* argument you must specify **1** for indirect Synchronization, **2** for direct Synchronization, or **3** for Internet Synchronization. The following is an example of calling the function to perform a one-way, indirect synchronization:

```
Call SynchronizeDBsJRO("C:\MyRepl.mdb","MySynchronizer",1,1)

Sub SynchronizeDBsJRO(strReplicaName As String, strSynchWith As
String, _
intSynchType As Integer, intSynchMode As Integer)

Dim rep As JRO.Replica
Set rep = New JRO.Replica
rep.ActiveConnection = strReplicaName
rep.Synchronize strSynchWith, intSynchType, intSynchMode
Set rep = Nothing

End Sub
```

Note that you can also use Microsoft Data Access Objects (DAO) code to synchronize two databases. The following code illustrates a direct synchronization that uses DAO:

```
Call SynchronizeDBs("C:\MyRepl.mdb", "L:\OtherRepl.mdb", 2)

Sub SynchronizeDBs(strDBName As String, strSyncTargetDB As String, _
intSync As Integer)

Dim dbs As DAO.Database

Set dbs = DBEngine(0).OpenDatabase(strDBName)

Select Case intSync
```

```
Case 1 'Synchronize replicas (bidirectional exchange).
dbs.Synchronize strSyncTargetDB, dbRepImpExpChanges
Case 2 'Synchronize replicas (Export changes).
dbs.Synchronize strSyncTargetDB, dbRepExportChanges
Case 3 'Synchronize replicas (Import changes).
dbs.Synchronize strSyncTargetDB, dbRepImportChanges
Case 4 'Synchronize replicas (Internet).
dbs.Synchronize strSyncTargetDB, dbRepSyncInternet
End Select

dbs.Close
End Sub
```

9. Can I use database-level passwords with replication?

The database password feature lets you set a single password on a database instead of having multiple logons with regular Microsoft Access user-level security. This feature is incompatible with replication. You cannot replicate a database that has a password set and you cannot set a database password on a replica. You can always implement user-level security in Microsoft Access. To do this, have all your users log on by using the same logon name and password. This has the same effect, is a more robust solution, and is more reliable than a database password. However, it does require that you distribute your secured workgroup file (System.mdw) with your replicas.

You can also create the effect of having a database password by creating a custom logon routine that is tied to an AutoExec macro or to a database Startup form.

10. How do I implement Access user-level security with replication?

Both replication and Microsoft Access user-level security are advanced features that require much advance planning to implement them successfully. You can combine the two, but you must consider how your secured replicated application will work. You can secure your application in the typical way, and any changes you make to the permissions of your database objects will be distributed during synchronization. However, you cannot synchronize workgroup files. Therefore, you must distribute your workgroup file (system.mdw) separately to each site that participates in replication. This can make administration difficult if you constantly have to add and remove users. Another option is to create common groups in different workgroup files that have identical names and PIDs. This enables each site to maintain its own workgroup file while still enabling access to replicated objects. For more information about how to set up Microsoft Access user-level security, see the *Access 2000 Security FAQ*:

> Microsoft Access Security FAQ available in download center
>
> http://support.microsoft.com/kb/207793

11. How do I update linked tables in a replicated database?

If you have code that deletes the linked tables and re-creates them, the code will fail in all replicas except for the Design Master. This behavior occurs because deleting linked tables and adding new tables are both considered design changes. To update the links, you must write code for your front-end replicated application that updates the **Connect** property of each attached **TableDef** object. The update to the **Connect** property must contain the *path/mdb* name of the back-end database. After you update to the **Connect** property, call the **RefreshLink** method for each **TableDef** object. This lets the Jet Database Engine know where the new data is located. This is not considered a design change for the purposes of replication. Therefore, the path will not propagate inappropriately to other replicas during synchronization. To update the link in all members of the replica set, change the replicable linked table to a local table in the Design Master, update the link, and change the table back to a replicable table.

12. How do I replicate an application database that has implemented split data and code MDBs?

You can replicate the application database, the table database, or both. However, you may find it more efficient to replicate only the table database. Although design changes are replicable, synchronizing them can cause project stability issues. There have been reported instances of design changes to forms and modules that do not propagate successfully to all replicas in a replica set. If it is possible, develop your application fully before you replicate and distribute it. If you have to make design changes later, distribute the front-end database separately from the tables. Replicating only the table database typically offers reduced synchronization times as an added benefit.

13. How do I compact a replicated database?

When you compact a replica in Microsoft Jet Database Engine 4.0, consider the following issues:

- After compacting and before synchronizing the replica with any other replica, make sure that the replica is copied back to the original *location/name*. If you do not do this, you will be adding another replica to the replica set. If the replica is the Design Master, it no longer will be the Design Master because it moved. The best approach is to make a backup of the replica before compacting, and then compact it to the same location/name.
- If you compact a replica that is corrupted, it will lose its replicable status and Design Master status if it is the Design Master. If you compact a corrupted replica, the replica returns to a typical non-replicated database. However, all the hidden system tables and fields are still present.

- For best results, compact the replica two times. The first compact performs the regular consolidation process and marks replicated objects that it has to delete. However, it does not actually delete these objects. The second compact deletes these objects and recovers space associated with the deleted replicated objects. Generally, this is only important in the Design Master. Ordinary replicas only have to compact one time.
- Although the previous advice on compaction is always important, it is very important in the Design Master, especially when you make design changes to Microsoft Access–specific objects. This is important because of how Microsoft Access "versions" its objects. If you make 80 changes to a form and you save it 80 times, there will be 80 copies of that form stored in your database so that the Jet Database Engine can apply those changes somewhere else. If you compact the database before you synchronize it, the Jet Database Engine will notice that 79 of those changes are irrelevant and do not have to be kept. However, if you synchronize first, you cannot reclaim the space and all 80 changes will have to be applied to every replica. Therefore, *always compact the Design Master two times before synchronizing*.

  You should compact replicated databases on a regular (daily or weekly) basis, because replication can frequently bloat the databases.

14. How do I repair a corrupted Design Master?

You can try to compact the corrupted Design Master by using the steps in How do I compact a replicated database? However, remember that if Jet has to repair any corruption in the database, the Design Master will lose its replicable status. To replace a corrupted Design Master, synchronize the surviving replicas and designate one replica as the new Design Master. Also see How do I recover a lost or corrupted Design Master?

15. What is Replication Manager?

Replication Manager is an optional tool that is included with Microsoft Office 2000 Developer Edition and Microsoft Office XP Developer Edition. This tool assists you in some aspects of replication. Replication Manager was not included with the Microsoft Visual Studio Tools for the Microsoft Office System 2003. You can use the Replication Manager that is included with Office 2000 Developer Edition and with Office XP Developer Edition with databases that were created in Access 2003. The Replication Manager that is included with the Microsoft Office 97 Developer Edition cannot handle Access 2000 or later databases. Replication Manager includes features that let you schedule and execute unattended synchronizations without programming. Additionally, it lets you use indirect synchronization and Internet synchronization, and provides tools for examining the synchronization history for a replica. One of the components of Replication

Manager is the Synchronizer. This component performs the unattended synchronizations that you set up with Replication Manager.

For more information about indirect synchronization, see What is indirect synchronization and how do I make it work?

16. My remote access connection for synchronization is really slow. How can I speed up the synchronization process?

Consider using indirect synchronization that is described in What is indirect synchronization and how do I make it work?

17. What is indirect synchronization and how do I make it work?

Indirect synchronization requires that you use Replication Manager or a third-party solution. Trigeminal Software has created the TSI Synchronizer which lets you perform indirect synchronizations programmatically. For more information, see the "Additional references and resources" section at the end of this document.

When you synchronize two replicas by using indirect synchronization, a Synchronizer on one computer exchanges changes with a Synchronizer on another computer. This avoids directly opening a database over a wide area network (WAN).

The following are benefits of using this process:

- It greatly reduces the chance of database corruption that could be caused by a line drop during a database update.
- If an indirect synchronization fails, Replication Manager will just re-send any changes on the next pass.

To use indirect synchronization, you will have to configure the Replication Manager at both the local and remote sites and create separate dropbox locations for each Replication Manager. Each Synchronizer must have its own dropbox. The Synchronizer will use the same dropbox for all the replica sets that it manages.

To do this, set up the dropboxes or shared folders. These will be used to store the changes. You may not want to locate the replicas in the shared folders, or indirect synchronization may be bypassed in favor of a direct synchronization.

Note *It is possible to store the replicas in shared folders. However, if the Synchronizer on the computer that you are trying an indirect synchronization with is not running, Jet will try the next synchronization method in the synchronization priority list. For more information, see What effect does the Synchronization priority list setting in the Replication Manager configuration have* *on indirect synchronizations?*

To configure Replication Manager on the local computer, follow these steps:

1. Install Replication Manager.
2. Create a shared folder. This will be the Synchronizer dropbox. We recommend that you use a top-level share, so that the path appears as follows: \\servername\share. If you are using Windows NT® or a later version of Windows, you can define the share as \\servername\share$. The dollar sign ($) at the end of the share name makes the share invisible to most users. Designating the share as hidden does not affect the functionality of the share in any way.
3. Configure Replication Manager for indirect synchronization by checking the option to support indirect synchronization. Follow the rest of the steps in the wizard by specifying the shared folder that you created as the dropbox, by naming the Synchronizer, and by selecting a synchronizer log file.
4. **Note** We highly recommend that you locate this dropbox on the same computer that its Synchronizer runs on.
5. Open Replication Manager, and select the replica to be managed.
6. Create a new unmanaged replica by using the replica that you created in step 4 as the source replica. This unmanaged replica will be distributed to the computer that you will be synchronizing with.
7. Optionally, set up a synchronization schedule.

To configure Replication Manager on the remote computer, follow these steps:

1. Install Replication Manager and create a share for the Synchronizer dropbox, as in steps 1 and 2 for the local computer.
2. Copy the replica that you created in step 5 from the local to the remote computer by using a regular file copy, and put it in a folder.
3. Open the newly copied replica in the folder on the client computer in Replication Manager, and then elect to manage the replica.
4. Connect back to the local computer and synchronize. This informs the local computer where the copied replica is on the remote computer.

Whenever the Replication Manager manages a replica, it writes the location of its dropbox into the replica. With remote connections, you must now notify other replicas of the dropbox location. That is why you must follow step 4. You notify other replicas of the new address or dropbox in one of the following two ways:

- Create a new replica from this managed replica.
- Synchronize with an existing replica.

18. What effect does the Synchronization priority list setting in the Replication Manager configuration have on indirect synchronizations?

When you execute an indirect synchronization by using the **Synchronize** method in the Microsoft Jet and Replication Objects library (JRO) or by clicking **Synchronize Now** in Replication Manager, a "you pick" synchronization will be executed by using the Synchronizer. This means the Synchronizer will first read the registry for the order that it should try to execute each type of synchronization. The default order is as follows:

1. file system (dropbox)
2. Internet
3. direct synchronization (If the first two methods fail.)

These attempts do not cause any significant performance hit on synchronizations. Users can control the order of the attempts and even disable a particular synchronization type by modifying registry values. Realize that if you try an indirect synchronization, and the Synchronizer that you are trying synchronization with is not running, Jet will move to the next method in the list. By default, Jet will next try Internet Synchronization if it is configured. If the Internet synchronization fails or if Internet Replication has not been configured, Jet will then try to perform a direct synchronization of the managed replicas. This attempt will be successful if the replicas are located in shared folders. Direct synchronizations are not recommended over an unstable network connection such as a dial-up-networking connection or a Wide Area Network connection.

If the direct synchronization also fails, the Jet Synchronizer will then write a message file to the dropbox of the other Synchronizer if it is online. This behavior causes an indirect synchronization when the Synchronizer on the remote computer is back online. If the dropbox is not online, Jet will add the requested indirect synchronization to the system tables of the requesting replica. The message file is then written to the remote dropbox when the remote dropbox is back online. This point is important to remember because even though it is possible to store the managed replicas in shared folders and have indirect synchronization occur, Jet will first try the other synchronization methods in the priority list. The other methods must all fail before Jet will go back to indirect synchronization. Only after these methods fail will Jet write the message file to the dropbox of the local Synchronizer, or schedule the request to perform the indirect synchronization when the remote Synchronizer is back online.

19. Can I use JRO to perform indirect synchronization?

Yes, you must use Microsoft Jet OLE DB Provider and Replication Objects (JRO) to perform an indirect synchronization programmatically. For more information, see How can I synchronize in a one-way direction? You can also handle Internet synchronization by using JRO when the Internet server is correctly configured. This is a special form of indirect synchronization. For more information, see How do I set up Internet replication? Note that you must have MOD installed and configured for both of these synchronization types.

20. How do I create a partial replica?

The easiest way to create a partial replica is to use the Partial Replica Wizard and experiment. To do this in Access, point to **Replication** on the **Tools** menu, and then click **Partial Replica Wizard**. You can also use JRO or DAO to create partial replicas. Additionally, if you want to create partial replicas that have more than one partial replica filter, you have to use either JRO or DAO. When you have created a new partial replica, do not delete the full replica that it was based on. The full replica acts similarly to a "backup" from before you started creating partial replicas. Partial replicas can only synchronize with full replicas. A partial replica cannot synchronize with another partial replica. The term "backup" is used cautiously here. Because a partial replica contains only a subset of the data, it should never be considered a full backup of the information.

21. The Partial Replica Wizard only lets me define a partial replica filter to one table. How do I create additional replica table filters?

To add a filter to more than one table in a partial replica table, use Microsoft Jet and Replication Objects (JRO) or DAO. (For more information, see Microsoft Visual Basic® Help. Search for "Create Partial Replicas" to see steps and code samples.) The three primary database methods that you will use in JRO when you work with partial replicas are **Append**, **Delete**, and **PopulatePartial**. You can only populate a partial replica from the Design Master when it is open exclusively. This is illustrated in the following code:

```
Public Sub PartialRep()

 Dim repPartial As JRO.Replica
 Set repPartial = New JRO.Replica
 ' PopulatePartial requires an exclusive connection to the database.
 repPartial.ActiveConnection = "Provider=Microsoft.Jet.OLEDB.4.0;" &
_
 "Data Source=C:\PrtlNorthwind.mdb;" & _
 "Mode=Share Exclusive"

 repPartial.PopulatePartial "C:\Northwind.mdb"

End Sub
```

You can also perform the same operation from DAO by using code that is similar to the following:

```
 Sub PopulatePartialDB
 Dim db As DAO.Database
 Dim StrFullDB As String
```

```
Set db = OpenDatabase("C:\MyApp.mdb", True)
StrFull = "k:\direct.mdb"

db.PopulatePartial StrFullDB
db.Close
End Sub
```

For more information, see the "Additional references and resources" section at the end of this document.

22. Can I make replication work from Microsoft SQL Server™and Microsoft Access?

In earlier versions of Microsoft SQL Server and of Microsoft Jet, data could be replicated to a Microsoft Jet database, but changes that were made in the Microsoft Jet database could not be used to update the Microsoft SQL Server database. With Microsoft Jet 4.0, replication in combination with Microsoft SQL Server 7.0 or Microsoft SQL Server 2000 and support for bidirectional replication between Microsoft Jet and SQL Server has been implemented. Changes that are made to data in a Microsoft SQL Server database can be replicated to a Microsoft Jet database, and changes to the data in a Microsoft Jet database can be synchronized to and reconciled with a SQL Server database.

A mixed replica set that contains both SQL Server replicas and Microsoft Jet replicas is required for this feature to work. To start creating this replica set, first create a SQL Server database. If you are starting with a Microsoft Jet database, you can use the Upsizing Wizard in Microsoft Access to create a SQL Server version of your database. When your SQL Server database is made replicable, you can start creating SQL Server or Microsoft Jet replicas.

The following are some limitations:

- Only data can be replicated between Microsoft Jet and Microsoft SQL Server. Microsoft Access objects such as forms, reports, data access pages, macros, and modules cannot be replicated to Microsoft SQL Server. These objects will continue to reside only in the Microsoft Jet database.
- The only topology that is supported in Microsoft Jet or Microsoft SQL Server replication is "hub and spoke." The Microsoft SQL Server database is always the hub. The Microsoft Jet replicas are the spokes and cannot synchronize with other Microsoft Jet replicas. These replicas can synchronize only with their Microsoft SQL Server hub database.

For more information about replicating data between Microsoft SQL Server and Microsoft Jet databases, see "Implementing Merge Replication to Access Subscribers" in the documentation that is provided with SQL Server.

23. Why does my replica report that it has expired? How can I fix it?

The replica set retention period setting controls the number of days that non-synchronized system data is retained in system tables. The retention period is established when the database is first made replicable. If you replicate the database by using Replication Manager or JRO, the default retention period is sixty days. If you replicate the database by using Microsoft Access or Briefcase replication, the default retention period is one thousand days. Old information is purged during a database compact and is also purged periodically if a Synchronizer is managing the replica. You can change the retention period in the Design Master by using Replication Manager or by setting the **RetentionPeriod** property in JRO. A replica set should have a large retention period if replicas do not synchronize frequently. However, if replicas synchronize frequently and you want to keep replica size smaller, specify a shorter retention period.

24. Can I create MDEs for use with Replication?

Yes. However, you cannot convert a replicated database to an MDE file. You can convert a database that is in MDE format to a replicated database. To convert a replicated database to MDE format, you must convert the database to a non-replicated database. For more information, see Can I use replication with Visual SourceSafe?

25. Can I import and export data from a replicated database?

Yes. Just use the standard import/export process. However, if you import into a non-Design Master replica, and you create a new table, you cannot replicate it. For a new table to be replicable, you must create it in the Design Master. Remember that the replication system fields will be included if you import a replicable table from a replica. To remove the replication system fields, you will have to run make-table queries.

26. Why do I receive a conflict when I synchronize my replicas?

This occurs because more than one replica has updated data in the same record since the last synchronization. This occurs in the same column of the same record if you have chosen column-level tracking. Using Jet 4.0 in Access 2000, Access 2002, or Access 2003 will handle data errors just like it handles conflicts. Therefore, other causes for the conflict could include violations to the primary key constrictions and newly created validation rules.

27. How can I avoid or at least minimize data conflicts?

The best way to minimize conflicts is to prevent them. A book could be written about this topic. However, the following are several examples of how to avoid or minimize conflicts:

- ***Add new records instead of editing existing records***. For example, when you track inventory, keep a single field that has the quantity instead of creating a log table that lists the change. Then when two people each remove some of the same item, you do not receive a conflict when the two people edit the same row.
- ***Create smarter synchronization schedules***. Generally, it is a good idea to set up your synchronization schedule based on the way your application works. If one replica contains data that another one will need immediately, your application should make the synchronization immediately. This avoids potential conflicts when someone else changes the same record before the first set of changes has propagated. This will also help you avoid angry customers who wonder why they just gave someone an address change and why they have to do it all over again for someone else!

28. How does Microsoft Access resolve synchronization conflicts?

In earlier versions, synchronization conflicts were resolved based on an algorithm where the most often-changed copy of a record won the conflict. This algorithm worked, but it was unsophisticated and confusing.

Microsoft Jet 4.0 introduces an algorithm whereby replicas in a replica set are assigned priorities and the highest priority replica wins in the case of a synchronization conflict. Where priorities are equal, the replica with the lowest replica ID wins.

Note *The priority-based conflict-resolution algorithm will work together with the corresponding Microsoft SQL Server capability when you use Microsoft Jet and Microsoft SQL Server replication together.*

Replicas are assigned a priority when the replica is created. This priority is a real number between 0 and 100, inclusive. At the time a database is made replicable, the priority of the Design Master is set to **90**. The default priority of additional replicas is 90 percent of the priority of the parent replica. You can easily specify another priority value by using the **Create Replica** dialog box in Access or by using the **CreateReplica** method in JRO. To specify that the priority of a replica is greater than the priority of the replica from which it is created, users must have database Administer permissions. When you use the file system to make a copy of a replica, a replica is created that has the same properties and priority as the source replica.

Note *All replicas that are upgraded from an earlier version of Microsoft Jet are given an identical priority value of* ***90****. Replicas created through the* ***MakeReplica*** *method in DAO have a default priority equal to 90 percent of the parent replica's priority.*

29. How do I prevent replication data errors?

With Microsoft Jet 4.0 replication, the events that cause synchronization conflicts and synchronization errors are both viewed as synchronization conflicts. A single mechanism is used to record and resolve these conflicts. This mechanism makes the resolution of such problems easier. However, because the errors are logged in this manner, you still have to resolve the conflicts. Minimizing the occurrence of these errors (conflicts) will simplify administration of the replica set. Consider the following factors to minimize design errors (conflicts):

- Make sure that there are no open tables on the replica when you initiate design changes to that replica.
- Do not use bound forms. This may hold a table open, and this behavior blocks successful replication.
- Synchronize all replicas in the replica set before you make any more restrictive changes to table-level validation (TLV) rules. After you have made the change, synchronize all replicas again as soon as possible. This minimizes the chance that in the interim someone has entered data that breaks the new rule.
- Use these rules for other schema changes that are more restrictive such as the following:
  - When you reduce the maximum number of characters that are allowed in a field.
  - When you change a relationship from unenforced to **Enforce Referential integrity**.
  - When you add a unique index.
  - When any other change is made where someone may be violating the new rule in a replica at the same time that you are adding the rule to the Design Master.
- Compact the Design Master two times before you initiate the synchronization.
- Use primary keys that are either difficult or impossible for users of two different replicas to duplicate. This will help avoid the "Duplicate primary key" data error.

30. Can I create a replica that prevents users from deleting records?

Replicas that do not allow for record deletions are a new feature starting in Jet 4.0 when it is used with Access 2000. Preventing record deletions is a great way to increase the robustness of your replicated applications by eliminating many of the conflicts caused by referential integrity issues. To create these types of replicas, point to **Replication** on the **Tools** menu, and then click **Create Replica**. When you are prompted for the location of the new replica, click to select the **Prevent deletes** check box. You can only create a replica where records cannot be deleted by using the Access menus.

Note *Records that are deleted from other members of the replica set are still deleted from the Prevent Delete replicas during synchronizations.*

31. When a data error exists, how can I remove it?

There are no data errors in Access 2000-2003/Jet 4.0 replication. There are only conflicts. You can resolve conflicts by using the following methods:

- Use the Microsoft Replication Conflict Viewer.
- Use your own custom replication conflict manager.
- Manually delete the records from the appropriate Conflict tables.
- Delete the Conflict tables. They will be re-created automatically the next time that a conflict occurs.

32. Is there any way to create my own replication conflict manager?

One technique that you can use in Jet 4.0 is to use the new MSysConflicts table. This table contains lots of information about conflicts. The following is a sample query that you could run that uses the MSysConflicts table and other replication system tables to return a list of all the tables that have conflicts. The query separates the tables into counts of the original conflicts and the conflicts that were related to the original ones through what would have otherwise been a synchronization error. (The Conflict Viewer also makes this differentiation.)

```
SELECT
 Max(tg.TableName) AS src_Table,
 Max(st.SideTable) AS conflict_table,
 Count(IIF(c_a.ConflictIdGuid <> c_a.s_Guid,Null, 1)) As CnfCount,
 Count(IIF(c_a.ConflictIdGuid = c_a.s_Guid, Null, 1)) As
CnfChildCount
FROM
 (((MSysTableGuids AS tg INNER JOIN
 MSysSideTables AS st ON tg.s_GUID = st.TableGuid) INNER
JOIN
 MSysConflicts AS c ON st.TableGuid = c.BaseTableGuid)
INNER JOIN
 MSysReplicas AS r ON c.WinningReplicaId = r.ReplicaId)
INNER JOIN
 MSysConflicts AS c_a ON c_a.ConflictIdGuid = c.s_Guid
WHERE
 c.ConflictIDGuid = c.s_Guid
GROUP BY
 tg.TableName;
```

To return a recordset that contains the winning and the losing records in a conflict, you can use a query this is similar to the following:

```
SELECT
 Customers.*,
 Customers_Conflict.*
FROM
 (Customers_Conflict INNER JOIN
 MSysConflicts ON Customers_Conflict.ConflictRowGuid =
MSysConflicts.s_GUID) LEFT JOIN
 Customers ON MSysConflicts.WinningRowGuid = Customers.s_GUID
```

This query uses special fields in the MSysConflicts table that determine the actual record that caused the conflict. This can be very helpful with a duplicate primary key conflict. This conflict consists of two records that have different **s_GUID** values.

For more information about the MSysConflicts table, see the "Additional references and resources" section.

Another technique is to write your own Visual Basic code to resolve the conflicts. There is sample code and more details about custom conflict manager routines in the following book:

> Getz, Ken, Paul Litwin, and Mike Gilbert. *Access 2000 Developer's Handbook, Volume 1: Desktop Edition*. Alameda, CA: SYBEX, 1999.

However, the following is a simple routine that uses the **ConflictTables** property in JRO to display the name of any table with a conflict in the Debug window:

```
Public Function Resolve() 'Find tables with conflicts
 Dim conn As New ADODB.Connection
 Dim rs As New ADODB.Recordset
 Dim repRW As New JRO.Replica

 conn.Open "Provider=Microsoft.Jet.OLEDB.4.0;" & _
 "Data source=C:\demo\NWindRW.mdb;"
 repRW.ActiveConnection = conn
 Set rs = repRW.ConflictTables
 rs.MoveFirst
 While NOT rs.EOF
 Debug.Print rs(0) & " has a conflict and"
 Debug.Print rs(1) & " is the name of the a conflict table."
 rs.MoveNext
 Wend
End Function
```

33. Will replication work with Banyan Vines and LANtastic?

These networks are currently not supported for Microsoft Access replication.

34. Troubleshooting tips for Novell networks.

One problem could be that the client computer is running Microsoft Windows and is using the Novell Client. If this is the case, load the Microsoft Client. On the network side, make sure that the number of record locks per connection is set to

10,000 and that the maximum number of record locks that the server can handle is set to **200,000**.

35. Why are some of my design changes not propagated to the replicas?

Make sure that the replicas are closed before you try to synchronize design changes. As with any synchronization, compact all the replicas two times before you synchronizing them. If the problem is that the code is not running, open the database by using the **/decompile** command-line switch, and then recompile all the code. However, the best resolution to this issue may be to create a split database application.

36. I am having frequent database corruption problems. What can I do?

Microsoft Access does not handle data collisions well. Make sure your network is not faulty. A faulty network will cause your database to crash and to become corrupted, or your database may be put in a “suspect” state. When a replica is corrupted, it cannot participate in synchronization.

Before you try to replicate design changes, make sure to synchronize all your other replicas first so that the data is consistent across all members of your replica set. Compact all members before synchronizing.

Another way to avoid frequent database corruption problems is to avoid creating relationships between your tables. Instead, rely on code and queries to maintain referential integrity. This can make resolving conflicts on unique indexes easier to deal with if you do not also have to resolve relationships.

If you are synchronizing over a slow WAN line, consider using indirect synchronization. For more information, see What is indirect synchronization and how do I make it work?

37. How do I remove a defunct (deleted) replica?

If you try to synchronize with a deleted replica from within Microsoft Access, and Access cannot find the replica, it will remove the database from the replica set. The replica should no longer appear on the Replication Manager map. Replication Manager examines the MSysReplicas table to create the map. You cannot open the MSysReplicas table yourself and delete records from it because it has read-only permissions that are set by Microsoft Access. Additionally, you can remove a deleted replica by using the **Remove Defunct Replica** command in Replication Manager.

38. Should I use Briefcase manager to handle replication?

We recommend that you use the Microsoft Access UI, Replication Manager, or you use Microsoft Visual Basic for Applications and JRO to handle replication. You can use Briefcase manager for simple replication, but it is limited in that you can only synchronize on demand in a bidirectional way. You cannot set up a schedule or synchronize one-way only.

39. Can I use replication with Visual SourceSafe?

Before you can replicate with Microsoft Visual SourceSafe®, you must remove the database from source code control. You can set properties such as **SetObjectReplicability** on objects that you do not want to be replicated. This makes it easy to replicate the database after it is removed from source code control.

40. How do I remove replication?

If you are just starting out and you have not made many changes to your data and objects, you probably created a backup copy of your original database. Search for the backup copy in the folder where the replicated database is. The backup copy will have the same base file name as the replicated database and will have a .bak extension. If you have proceeded beyond that point, follow these steps:

1. Create a new database and import all the objects from the replicated database, except tables.
2. Close the new database and open the replicated database. Create a new query and select the first table in the **Show Table** dialog box. Add all the fields except for the replication fields (**s_Generation**, **s_Guid**, **s_Lineage**, **s_ColLineage**), unless they are used in your application. If they are used in your application, add them.
3. On the **Query** menu, click **Make Table Query**. Type the name of the new table, click the **Another Database** option, and then type the name of the database that you just created in step 1.
4. Run the query. Repeat this process for every table in the database.
5. Close the replicated database and open the new database from step 1. Re-create all indexes, properties, and relationships that existed in the original replicated database.
6. Make sure to compact, and then repair the new database.

If you do not want to re-create all the properties and indexes on your tables and you do not mind having the **s_Guid** field, but you want the other fields to be removed, you can do the following:

1. Delete all the relationships in your database or make them unenforced by clearing the **Enforce Referential Integrity** check box. You must perform this step because the Jet Database Engine will not allow a relationship that enforces referential integrity between a replicated table and a local table.
2. Right-click each replicated table, click **Properties**, and then click to clear the **Replicated** check box.
3. Import the tables from the replicated database as you did for the other objects.
4. Re-create the relationships in your database, or change their type back to **Enforce Referential Integrity**. All the properties such as **InputMask**, **Format**, and **Caption**, and all the indexes will still be there.

For another method that you can use to un-replicate a Jet 4.0 database in Access 2000, Access 2002, or Access 2003 see the "Additional references and resources" section at the end of this document.

41. How do I implement a progress meter to display synchronization status?

This property is not exposed to developers in JRO or DAO. It does exist in Replication Manager and in the Access UI. However, neither one can be manipulated with your code very effectively.

You can use a third-party solution. Trigeminal Software has created the TSI Synchronizer which provides a method to expose the synchronization progress. For more information, see the "Additional references and resources" section at the end of this document.

42. Should I use backup utilities with my replicas?

You do not have to. Replication itself is a good mechanism for creating backups. Use another replica member on another physical drive or on another computer to back up your replicas. Synchronize regularly to make sure that there is a minimum amount of downtime in case you ever have to restore from the backup replica. Generally, it is a good idea to consider how much time that you or your customer can afford to take to re-create the data that is lost. That answer helps determine the minimum synchronization interval.

If a replica member is corrupted or lost because of media theft or failure, create a new replica from another replica in the replica set. If you lose the Design Master, see How do I recover a lost or corrupted Design Master?

43. Can I replicate through e-mail instead of over a network or dial-up connection?

No. This feature is currently not supported.

44. How do I recover a lost or corrupted Design Master?

Synchronize all the replicas in your replica set to make sure that all existing members are up to date. Then, choose one of the members as the new Design Master. To do this, use the **Replication** menus, use the Replication Manager, or use code by setting the **DesignMasterID** property.

45. How can I transfer Design Master status to another replica member?

Use Microsoft Access to transfer the Design Master status. To do this, follow these steps:

1. On the **Tools** menu, point to **Replication**, and then click **Recover Design Master**.
2. In the **Synchronize Database <*dbname*>** dialog box, click the **Make '<*path\filename*>' the Design Master** check box.

Behind the scenes, the Jet Database Engine is transferring the **DesignMasterID** property to the **ReplicaID** property of the target replica. The following JRO code example will work in Microsoft Access:

```
Sub SetNewDesignMaster(repPresentDM As String, repNewDM As String)
 Dim rep1 As JRO.Replica
 Dim rep2 As JRO.Replica
 Set rep1 = New JRO.Replica
 Set rep2 = New JRO.Replica
 ' Open current Design Master exclusive
 rep1.ActiveConnection = "Provider=Microsoft.Jet.OLEDB.4.0;" & _
 "Data Source=" & repPresentDM & ";" & _
 "Mode=Share Exclusive"
 ' Open full replica that will become Design Master
 rep2.ActiveConnection = repNewDM
 rep1.DesignMasterId = rep2.ReplicaId
 rep1.Synchronize repNewDM, _
 jrSyncTypeImpExp, jrSyncModeDirect

End Sub
```

To perform the same task by using DAO, use code that is similar to the following:

```
Sub SetNewDesignMaster(stroldDM, strNewDM)
Dim dbs As DAO.Database
Dim newdmdb As DAO.Database

' Open current Design Master in exclusive mode.
Set dbs = OpenDatabase(stroldDM, True)
' Open database that will become the new Design Master.
Set newdmdb = OpenDatabase(strNewDM)
```

```
dbs.DesignMasterID = newdmdb.ReplicaID
dbs.Synchronize strNewDM, dbRepImpExpChanges
dbs.Close
newdmdb.Close
End Sub
```

Consider the following points:

- A Design Master can set one of its replicas to be the new Design Master. A replica can make itself the new Design Master. However, it cannot make a different replica the Design Master.
- Under extreme circumstances you can set this property in the current replica. An extreme circumstance might be if the Design Master is erased or corrupted. However, setting this property at a replica when there is already another Design Master in the set might partition your replica set into two irreconcilable sets. This behavior prevents any additional synchronization of data.

46. Can I have more than one Design Master in a replica set?

This is rarely a good idea. An exception to this is might be to transfer the Design Master status to another replica member for a specific developer. Design changes should only be made in one location. If you have to transfer the Design Master status to another replica member, see How can I transfer Design Master status to another replica member?

Some people have tried to use replication as a limited form of version control. Although it does not keep historical information, this does let you limit changes to one developer at a time. By “checking out” the database, the Design Master status is transferred to that replica. However, this technique is not a good idea unless you synchronize frequently. It is not a good idea because some replicas may not recognize which replica is the Design Master, and may not recognize how to apply schema changes that the different Design Masters have made.

47. How can I detect that the current replica member is the Design Master?

If the **ReplicaID** is equal to the **DesignMasterID**, the database name that is passed as an argument to this function is the Design Master. This is illustrated in the following code:

```
Function IsDesignMaster(strDB As String) As Boolean
 Dim rep As JRO.Replica
 Set rep = New JRO.Replica
 ' Open Database passed as argument
 rep.ActiveConnection = strDB
 ' Initialize return code to False
 IsDesignMaster = False

 If Len(rep.ReplicaId & "") > 0 Then
 If rep.DesignMasterId = rep.ReplicaId Then
 IsDesignMaster = True
 End If
 End If
End Function
```

To perform the same action by using DAO, you can use code that is similar to the following:

```
Public Function IsDesignMaster(strDBName) As Boolean
Dim dbs As DAO.Database
' Initialize return code to False
IsDesignMaster = False
' Open Database passed as argument
Set dbs = OpenDatabase(strDBName)

If Len(dbs.ReplicaID & "") > 0 Then
If dbs.DesignMasterID = dbs.ReplicaID Then
IsDesignMaster = True
End If
End If
End Function
```

48. Why should I not use replication for transactional processing applications?

Transactions are typically defined as a "unit of work." An example of this is in a banking application where an individual transfers $100.00 from a savings account to a checking account. The bank would only want the transaction to succeed if the amount was withdrawn from the savings account and deposited in the checking account. From a programmatic point of view, the customer’s savings table would need an SQL Update statement and the checking table would need a separate SQL Update statement. Imagine the customer’s dismay if $100.00 were debited from the savings account and not credited to the checking account! This transaction should succeed on an all-or-nothing basis. In other words, the updates to two separate tables should either completely succeed or completely fail.

In a banking environment, there is typically one central database against which all transactions such as updates and deletes are processed. This ensures data consistency and integrity.

Microsoft Access Replication was not designed to be used with transactions because replication is not a transaction-based system. Individual transactions that are run on any one replica will work as a transaction only for that replica. Only the results of the transaction are replicated to other replica members in the replica set. The heart of the problem is maintaining transactional consistency between replica members.

To illustrate the replication consistency problem, consider the following two transactions performed on the same customer from two different replicas.

	Replica A	Replica B
Starting combined balance	$100	$100
Savings	($20)	-
Checking	-	($20)
Ending combined balance	$80	$80

There would be a conflict record of the ending combined balance based on the results of the transaction. Each replica reports the correct balance, generated by a different transaction. Neither the Savings debit in Replica A nor the Checking debit in Replica B would generate a conflict record because these updates were performed on different tables.

If you are looking for absolute transaction consistency, consider using a product such as Microsoft SQL Server. SQL Server offers true transaction journaling, better recovery options if a crash occurs, rollback capability, better handling of potential deadlock situations, and superior multiuser capabilities.

49. How do I set up Internet replication?

This is a complex process, and it has many steps. For more information about how to do this, see the following Microsoft Developer Network (MSDN®) Web site:

> http://msdn.microsoft.com/library/backgrnd/html/intrjet4.htm

For more information, see the following book:

> Getz, Ken, Paul Litwin, and Mike Gilbert. *Access 2000 Developer's Handbook, Volume 1: Desktop Edition*. Alameda, CA: SYBEX, 1999.

50. Can I use Replication Manager to schedule synchronizations over the Internet?

No, you cannot use Replication Manager to do this. This feature is not supported for Internet synchronizations. However, you can perform Internet synchronizations by using Visual Basic for Applications and JRO or by using Visual Basic for Applications and DAO. This is discussed in the next question.

51. How can I synchronize over the Internet by using Visual Basic for Applications code?

The JRO **Synchronize** method uses the following syntax:

```
Replica.Synchronize(Target [, SyncType] [, SyncMode])
```

When you synchronize replicas over a local area network, you must specify the local area network path of the replica that you want to synchronize with for the *Target* argument. When you synchronize replicas over the Internet, you must specify the URL address of the Internet server for the *Target* argument instead of specifying a local area network path. Additionally, you must specify the **jrSyncModeInternet** constant for the *SyncMode* argument.

When you supply the URL address of the Internet server, your code does not have to supply the full path of the replica on the server. For example, if your Internet server name is *MyServer* and it contains a replica named Northwind.mdb in a shared Scripts folder, you would use the following syntax:

```
Sub InternetSync()
Dim rep As JRO.Replica
Set rep = New JRO.Replica
rep.ActiveConnection = "C:\Program Files\Microsoft Office\" & _
"Office\Samples\Northwind.mdb"
rep.Synchronize "http://MY_SERVER", jrSyncTypeImpExp, _
jrSyncModeInternet
Set rep = Nothing
End Sub
```

If the database on the remote computer is secured, you have to add the UserID and Password information and the location of the workgroup file to the Active Connection. The following is an example:

```
rep.ActiveConnection = _
"Data Source=C:\Program Files\Microsoft Office\" & _
"Office\Samples\Northwind.mdb;" & _
"Password=test;User ID=test;" & _
"Jet OLEDB:System database=C:\Secured.mdw"
```

To do the same type of synchronization by using DAO, you can use the **Synchronize** method off the **DAO Database** object. The following example uses the Microsoft Access **CurrentDb** function:

```
 Sub SyncReplicas()
 Dim db As DAO.Database
 Set db = CurrentDb()
 db.Synchronize "http://MyServer", dbRepSyncInternet +
dbRepImpExpChanges
 End Sub
```

52. Do I have to install Replication Manager on the client computers for them to perform an Internet Synchronization?

No. It is only necessary to have Replication Manager and a Synchronizer on the client computers when you use indirect replication.

53. When I try to synchronize my Anonymous replica over the Internet, I receive the “The search key was not found in any record” error message on the client computer.

This is a known bug. To resolve this issue, install the latest service pack for the Microsoft Jet Database Engine version 4.0 and the latest update of the Jet Replication files. You can download these updated files from the following Microsoft Knowledge Base articles:

How to obtain the latest service pack for the Microsoft Jet 4.0 Database Engine

http://support.microsoft.com/kb/239114

Updated version of the Microsoft Jet 4.0 replication files is available in the Download Center

http://support.microsoft.com/kb/320805

54. How do I convert a Microsoft Access 97 replica set to Microsoft Access 2000?

1. In Access 97, make sure that the replica set has a Design Master. If it does not, choose a member replica, and then promote it to the Design Master.
2. Synchronize the Design Master through all members of the set two times to make sure that all members of the set have the same data and design.
3. Compact all members of the replica set.
4. Convert all members of the replica set to Access 2000 (Jet 4.0).

   Note All Access objects in a newly converted replica will appear un-replicated (there will be no replicated icon on the object in the Database window) until the newly converted replica is synchronized with the converted Design Master of the replica set.

5. Synchronize all members of the replica set to the converted Design Master.

Important*: There is a known issue when you convert Access 97 partial replicas to Access 2000. When you synchronize a converted partial replica with a converted Access 2000 Design Master, all the Access objects in the converted Access 2000 partial replica are deleted. Currently, the only resolution is to create new partial replicas in Access 2000.*

There are additional issues to consider when you convert replica sets that use indirect replication and Internet replication. These issues are outlined in detail in the Microsoft Office 2000 Developer Readme file. The ReadmeODE.htm file is located in the root directory of the Microsoft Office 2000 Developer CD.

55. Can I convert my split Access 97 application's front-end database to Access 2000, but leave the replicated back-end database in Access 97?

No, this is not possible. You can create a linked table in an Access 2000 database that points to an Access 97 table. However, the data will be read-only. If you try to modify the data, you receive the following error message:

> Operation not supported on replicable databases that have not been converted to the current version.

This behavior is by design. It is caused by the tracking layer of the Microsoft Jet Database Engine. To work around this problem, convert the back-end database, or leave the front-end database in Access 97 format.

56. If I have to use Replication Manager and a Synchronizer for indirect synchronization, do I have to purchase a copy of Microsoft Office Developer for each user?

No, the license for Microsoft Office Developer lets you freely distribute all the necessary replication components. For more information, see the following Microsoft Knowledge Base article:

> How to distribute Replication Manager 4.0 by using the Package and Deployment Wizard
>
> http://support.microsoft.com/kb/240967

FOR MORE INFORMATION

Additional references and resources

- Getz, Ken, Paul Litwin, and Mike Gilbert. "Mastering Replication." Chapter 9 in *Access 2000 Developer's Handbook, Volume 1: Desktop Edition*. Alameda, CA: SYBEX, 1999.
- Getz, Ken, Paul Litwin, and Mike Gunderloy. "Mastering Replication." Chapter 9 in *Access 2002 Enterprise Developer's Handbook*. Alameda, CA: SYBEX, 2002.
- Microsoft Knowledge Base articles

  http://support.microsoft.com/search
- Database replication in Microsoft Jet 4.0 white paper

  http://msdn.microsoft.com/library/techart/dbrepjet.htm
- Internet synchronization with the Microsoft Jet Database Engine: A technical overview

  http://msdn.microsoft.com/library/backgrnd/html/intrjet4.htm

The Web sites listed here are external to Microsoft. Use them at your own risk.

- Trigeminal Software

  http://www.trigeminal.com

  This Web site contains many replication resources. Trigeminal provides several free Jet replication tools and utilities. This includes the TSI Access 2000 Un-Replicator, the TSI Synchronizer, and TSI Replication System Fields Removal Utility. Many of these utilities exist in Jet 4.0 and Jet 3.5 versions.
- *Access, Visual Basic, and Office News*, published monthly by Advisor Publications.

  For subscription information, call (800) 336-6060, or visit the following Advisor Web site:

  http://www.advisor.com
- *SmartAccess* newsletter, published monthly by Pinnacle Publishing.

  For subscription information, call (800) 788-1900, or visit the following Pinnacle Web site:

  http://www.pinpub.com