

## Microsoft CRM Integration: Siebel Email Attachments Import - C# and MS Trans...

---

Microsoft CRM - CRM answer from Microsoft Business Solutions has aggressive pricing, going down and making it affordable for small companies. We see cases when Microsoft CRM replaces such traditional CRM systems as Siebel. It is not necessary, that clients decided to replace it themselves - they may be victims of their systems - the example: Great Plains had alliance with Siebel several years ago and we saw multiple clients with Great Plains - Siebel tandem. Now Great Plains integrates with MS CRM.

This article is for programmer, who needs data to be migrated from Siebel or other CRM to MS CRM.

Today's topic is Siebel emails, stored in the files, import into MS CRM database. Each message exists in the form of separate file in the import directory. We will use custom SQL table, created in MS SQL Server 2000:

```
if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[CrmAttachImporter]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)
```

```
drop table [dbo].[CrmAttachImporter]
```

```
GO
```

```
CREATE TABLE [dbo].[CrmAttachImporter] (
```

```
[Id] uniqueidentifier ROWGUIDCOL NOT NULL ,
```

```
[CrmActivityGuid] [varchar] (50) NOT NULL ,
```

```
[MsgFileName] [varchar] (2000) NOT NULL
```

```
) ON [PRIMARY]
```

```
GO
```

Comments to this table - its goal is storing MS CRM activity GUID relationship to file name with email attachment (email message), which needs to be attached to the activity. We will store activity GUID in the field CrmActivityGuid and file name in the import directory of the attachment in the MsgFileName field.

Configuration file for our utility will be the following:

```
provider=SQLOLEDB;Initial Catalog=Albaspectrum;Data Source=MSSQL1;User Id=sa;Password=sa;
```

```
data
```

```
CrmAttachImporter
```

```
CrmActivityGuid
```

```
MsgFileName
```

Here we described MS SQL Server connection string, the path to messages-files in the file system, table name, which stores the relations Activity GUID and file names, table column names, required for import procedure

To control import process we will use free logging library for .NET: log4net. You can get it here:

<http://logging.apache.org/log4net/>

Let's look at the method of potential attachments catalog scanning:

```
public void scanFolder() {
```

```
log = LogManager.GetLogger(typeof(AttachImporter));
```

```
DOMConfigurator.Configure(new FileInfo("log.config"));
```

```
try {
```

```
DirectoryInfo dirInfo = new DirectoryInfo(msgFolder);
```

```
FileInfo[] files = dirInfo.GetFiles();
```

```
Hashtable emails = new Hashtable();
```

```

foreach (FileInfo fileInfo in files) {
log.Debug("Analizing file: " + fileInfo.Name);

Guid activityId = GetActivityIdByFileName(fileInfo.Name);

if (! activityId.ToString().Equals(new Guid().ToString())) {
emails.Add(activityId, fileInfo.DirectoryName + @"\" + fileInfo.Name);

Console.WriteLine("Marked for import: " + fileInfo.Name);
log.Debug("Marked for import: " + fileInfo.Name);
}
else {
Console.WriteLine("Not found in activity import list: " + fileInfo.Name);
log.Debug("Not found in activity import list: " + fileInfo.Name);
}
}

ProcessMessages(emails);
}
catch (Exception e) {
Console.WriteLine(e.Message + "\r\n" + e.StackTrace);
}
}

```

Central place in this method is checking on the relationship existence in the import table for CRM Activity GUID and file name in the import directory:

```

private Guid GetActivityIdByFileName(string fileName) {
//create the database connection
OleDbConnection conn = new OleDbConnection(connectionString);

conn.Open();

//create the command object and store the sql query
OleDbCommand command = conn.CreateCommand();

command.CommandText = "SELECT " + activityGuidColumn + ", " + msgFileNameColumn + " FROM " +
tableName + " WHERE UPPER(LTRIM(RTRIM(" + msgFileNameColumn + "))) = UPPER(LTRIM(RTRIM(?)))";

log.Debug("Preview checking SQL query: " + command.CommandText);
log.Debug("Using file name: " + fileName);

command.Parameters.Add(new OleDbParameter(msgFileNameColumn, fileName));

//create the datareader object to connect to table
OleDbDataReader reader = command.ExecuteReader();

if (reader.Read()) {
Guid activityGuid = new Guid(reader.GetString(0));

reader.Close();
conn.Close();

return activityGuid;
}
else {
reader.Close();
conn.Close();
}
}

```

```
return new Guid();
}
}
```

Importing messages cache is transferred as a parameter to the method, which does attachment import into MS CRM:

```
private void ProcessMessages(Hashtable emails)
{
try
{
log.Debug("Start importing process");
```

```
CRMConnector crmConnector = new CRMConnector();
```

```
// Connect to CRM DB
crmConnector.SetCrmConfigPath(Environment.SystemDirectory + "/Albaspectrum/MSCRMGateway.xml");
crmConnector.SetCrmContentType(Environment.SystemDirectory + "/Albaspectrum/ContentType.txt");
crmConnector.Connect(log);
```

```
if (emails != null) {
ICollection keys = emails.Keys;
int attCounter = 0;
```

```
foreach (Guid o in keys) {
string attName = (string)(emails[o]);
```

```
crmConnector.AddAttachmentToActivity(o, attName, (new FileInfo(attName)).Length, attCounter);
```

```
attCounter++;
}
}
```

```
crmConnector.Close();
}
catch (Exception ex)
{
log.Debug(ex.Message + "\n" + ex.StackTrace);
}
}
```

All the required classes for working with MS CRM Activity are stored in MSCRMGateway library and described in these articles:

- [http://www.albaspectrum.com/Customizations\\_Whitepapers/Dexterity\\_SQL\\_VBA\\_Crystal/MSCRMCustomizationEmailAttachment.htm](http://www.albaspectrum.com/Customizations_Whitepapers/Dexterity_SQL_VBA_Crystal/MSCRMCustomizationEmailAttachment.htm)
- [http://www.albaspectrum.com/Customizations\\_Whitepapers/Dexterity\\_SQL\\_VBA\\_Crystal/MSCRMCustomizationClosedActivity.htm](http://www.albaspectrum.com/Customizations_Whitepapers/Dexterity_SQL_VBA_Crystal/MSCRMCustomizationClosedActivity.htm)

Happy converting and importing! If you would like us to do the job, give us a call 1-630.961.5918 or 1-866.528.0577 [help@albaspectrum.com](mailto:help@albaspectrum.com)

### **Short note about the author**

Boris Makushkin is Lead Software Developer in Alba Spectrum Technologies ? USA nationwide Microsoft CRM, Microsoft Great Plains customization company, serving Chicago, California, Arizona, Colorado, New York, Texas,

Georgia, Florida, Canada, Australia, UK, Russia, Europe and internationally ( <http://www.albaspectrum.com> ), he is Microsoft CRM SDK, Navision, C#, VB.Net, SQL, Oracle, Unix developer.

For reliable email consider looking into [hosted exchange](#) email services that can help any [business](#) prosper.

---

Author: Boris Makushkin

Article downloaded from page [eioba.com](#)