



# GENETRIX

SFMC + AZURE DATA FACTORY • INTEGRATION GUIDE

## Azure Data Factory + SFMC 2026: Pull Subscribers, Contacts & Email KPIs

---

*The official ADF SFMC connector cannot see Data Views, contacts, or custom DEs — and it is at End of Support. Here is the production-grade SFMC-to-SFTP-to-ADF pattern that actually works.*

by **Genetrix Technology**

Published June 10, 2026 • [genetrix.tech/blogs](https://genetrix.tech/blogs)

# Azure Data Factory + SFMC 2026: Pull Subscribers, Contacts & Email KPIs

By Genetrix Marketing • Published June 10,

2026 • <https://genetrix.tech/blogs/azure-data-factory-salesforce-marketing-cloud-2026-how-to-pull-subscribers-conta/>

*The official ADF SFMC connector cannot see Data Views, contacts, or custom DEs — and it is at End of Support. Here is the production-grade SFMC-to-SFTP-to-ADF pattern that actually works.*

If you are trying to pull subscriber data, contact records, and email engagement KPIs — sends, opens, clicks, bounces — from Salesforce Marketing Cloud into Azure Data Factory, you have probably run into the same wall every SFMC and Azure team hits eventually.

You set up the official SFMC connector in ADF. You install the package, assign scopes, configure OAuth. Then the dataset dropdown shows cryptic objects like `SFMC.Contacts_Schema` and `SFMC.TransactionalmessagingEmailNotSent`. No `_Open`. No `_Click`. No `_Sent`. No subscriber table. No custom Data Extensions.

This guide explains exactly why that happens and walks through the production-grade workaround that Genetrix and virtually every serious SFMC-Azure team is running in 2026. No hacks, no third-party middleware — just native tooling used correctly.

**End of Support — act now:** The official Salesforce Marketing Cloud connector for Azure Data Factory is currently at End of Support stage per Microsoft's own documentation. Microsoft is recommending migration to the ODBC connector. This makes the SFTP workaround in this guide not just a practical solution but the forward-compatible one.

## Why the Official ADF SFMC Connector Falls Short

The ADF SFMC connector is built on SFMC's REST API. For certain standard transactional objects, it works. But the data you actually need for subscriber analytics and email reporting does not live in those REST API objects. It lives in **Data Views** and **Data Extensions**, neither of which the connector exposes. This is a hard architectural limitation confirmed in Microsoft's official documentation. The connector does **not** support:

- Data Views (`_Sent`, `_Open`, `_Click`, `_Bounce`, `_Unsubscribe`)
- The Contacts and Subscribers table
- Custom Data Extensions of any kind
- Views and custom objects in general

## The Proven Workaround: SFMC to SFTP to ADF

The pattern that works in production is straightforward: use SFMC's Automation Studio to export the data you need into CSV files, push those files to SFMC's Enhanced FTP, then use ADF's SFTP connector to pick them up reliably on a schedule. This gives you complete access to every Data View and subscriber field with no custom code.

```

Step 1 - SFMC: Query Activity
-> Step 2 - SFMC: Data Extract
-> Step 3 - Bridge: Enhanced FTP (SFTP)
-> Step 4 - Azure: ADF SFTP Connector
-> Step 5 - Sink: Data Lake / Synapse / SQL

```

## Step-by-Step Implementation

### 1 — Build the Query Activity in SFMC

In Automation Studio, create a Query Activity that pulls the fields you need from the relevant Data Views into a staging Data Extension.

#### SFMC Query Activity – Email KPI Export SQL

```

SELECT
    s.SubscriberKey,
    s.EmailAddress,
    s.JobID,
    s.EventDate          AS SendDate,
    o.EventDate          AS OpenDate,
    c.EventDate          AS ClickDate,
    c.URL                AS ClickedURL,
    b.BounceCategory,
    b.BounceSubcategory,
    b.EventDate          AS BounceDate,
    u.EventDate          AS UnsubscribeDate

FROM _Sent              s
LEFT JOIN _Open         o ON s.SubscriberKey = o.SubscriberKey AND s.JobID = o.JobID
LEFT JOIN _Click        c ON s.SubscriberKey = c.SubscriberKey AND s.JobID = c.JobID
LEFT JOIN _Bounce       b ON s.SubscriberKey = b.SubscriberKey AND s.JobID = b.JobID
LEFT JOIN _Unsubscribe u ON s.SubscriberKey = u.SubscriberKey

WHERE s.EventDate >= DATEADD(day, -1, GETDATE())

```

**Important:** For large-volume orgs, always use incremental date filtering in your query. Pulling all-time data from `_Open` or `_Click` without a date filter can time out or produce files too large to process reliably.

### 2 — Add a Data Extract Activity and Schedule the Automation

After your Query Activity runs and populates the staging DE, add a **Data Extract** activity. Set the extract type to Data Extension Extract, point it at your staging DE, and configure the output as a CSV file dropped into your SFMC Enhanced FTP folder.

Wrap both activities inside an Automation with a schedule that matches your ADF pipeline cadence. If ADF runs at 4 AM, schedule SFMC's automation for 3 AM so the file is always ready before ADF picks it up.

### 3 — Configure the SFTP Linked Service with SSH Public Key Auth

In ADF, create a new Linked Service and select the SFTP connector. SSH public key authentication is the most secure and most reliable method.

Property	Value / Notes	Status
authenticationType	SshPublicKey	Required
host	Your SFMC Enhanced FTP hostname (from SFMC Setup)	Required
port	22 (standard SFTP)	Required

userName	Your SFMC SFTP username	Required
privateKeyContent	Base64-encoded OpenSSH private key. Store in Azure Key Vault — never hardcode.	Required
passPhrase	Only needed if your private key is password-protected	Optional

**Key format matters:** The ADF SFTP connector requires keys in OpenSSH format. If your key is a .ppk file generated by PuTTY, convert it to OpenSSH format first using PuTTYgen (Conversions menu) or the ssh-keygen -p -m OpenSSH command before uploading.

#### ADF Linked Service JSON – SFTP with SSH Public Key

```
{
  "name": "LS_SFMC_SFTP",
  "type": "LinkedService",
  "properties": {
    "type": "Sftp",
    "typeProperties": {
      "host": "YOUR_SFMC_SFTP_HOST",
      "port": 22,
      "authenticationType": "SshPublicKey",
      "userName": "YOUR_SFTP_USERNAME",
      "privateKeyContent": {
        "type": "AzureKeyVaultSecret",
        "store": { "referenceName": "AKV_LinkedService", "type": "LinkedServiceReference"
      },
      "secretName": "sfmc-sftp-private-key"
    },
    "skipHostKeyValidation": false
  }
}
```

## 4 — Build the ADF Copy Activity Pipeline

Create a pipeline with a Copy Activity. Set the source to your SFTP dataset, pointing at your SFMC Enhanced FTP folder with a wildcard file filter such as EmailKPIs\_\*.csv. Set the sink to Azure Data Lake Gen2, Azure Synapse, or SQL Database depending on your downstream use case.

Schedule the pipeline trigger to run after your SFMC Automation Studio job completes. For more robust orchestration, use an event-based trigger on the storage account if your SFTP files land in Azure Blob first.

### 2026 Setup Checklist

- Do not use the native ADF SFMC connector for Data Views, contacts, or custom DEs. It does not support them and is at End of Support.
- Use Automation Studio: Query Activity to stage a DE, then Data Extract to drop CSV on Enhanced FTP.
- Always use Enhanced FTP (SFTP), not the legacy FTP. SSH key auth only in production.
- Convert private keys to OpenSSH format before uploading to ADF. PuTTY .ppk files will not work.
- Store all credentials in Azure Key Vault. Never hardcode in Linked Service JSON.
- Add incremental date filtering to your SFMC query. Do not pull all-time Data View data on every run.
- Schedule ADF to run at least 30 minutes after your SFMC automation is expected to complete.
- Test end to end with a single Data View first (\_Sent is a good starting point) before expanding to the full schema.

## Frequently Asked Questions

### **Can I query Data Views directly with the ADF SFMC connector?**

No. Microsoft explicitly states in their documentation that the connector does not support retrieving views, custom objects, or custom Data Extensions. The connector is also now at End of Support stage. The SFTP export pattern described in this guide is the correct long-term solution.

### **Why SSH public key authentication over username and password?**

SSH public key auth is significantly more secure and operationally easier to manage. Passwords can expire or be rotated by another team member and silently break your pipeline overnight. Key-based auth is not subject to password policies, integrates cleanly with Azure Key Vault, and is the approach recommended in Microsoft's official ADF SFTP documentation.

### **How do I handle large volumes of tracking data without timeouts?**

Use incremental extraction — filter your SFMC query by EventDate to pull only the last 24 hours or 7 days depending on your cadence. For very high-volume orgs, also consider partitioned extracts where you split by JobID ranges or date segments and run multiple smaller exports in parallel through separate Automation Studio automations.

### **My private key was generated by PuTTY. How do I convert it?**

Open PuTTYgen, load your .ppk file, then go to Conversions › Export OpenSSH key. Save the file with a .pem extension. Alternatively, run `ssh-keygen -p -m OpenSSH -f your-key.ppk` in a terminal. The resulting file should start with `-----BEGIN OPENSSH PRIVATE KEY-----`.

## **Building a Salesforce to Azure Data Pipeline?**

Genetrix has run this exact integration pattern across enterprise and ISV environments — SFMC to SFTP to ADF to Synapse. If you are designing this architecture or untangling an existing one, our team can get you to production faster.

**Talk to Genetrix »**