MARKETING CLOUD ON CORE • MIGRATION GUIDE

Developer’s Translation Matrix: Migrating Legacy SSJS & AMPScript from MCE to Salesforce Core

Lift-and-shift fails on Marketing Cloud Core. Download the Genetrix Developer's Translation Matrix — a 10-point cheat sheet mapping every major MCE SSJS and AMPScript function to its Salesforce Flow, Apex, and Data Cloud equivalent.

by Genetrix Technology

Published June 10, 2026 • genetrix.tech/blogs

Genetrix Consulting LLC
304 S. Jones Blvd #5631, Las Vegas, NV 89107, USA | genetrix.tech

Developer’s Translation Matrix: Migrating Legacy SSJS & AMPScript from MCE to Salesforce Core

By Genetrix Marketing • Published June 10, 2026 • https://genetrix.tech/blogs/developers-translation-matrix-migrating-legacy-ssjs-ampscript-from-marketing-cloud-engagement-to-salesforce-core/

Lift-and-shift fails on Marketing Cloud Core. Download the Genetrix Developer's Translation Matrix — a 10-point cheat sheet mapping every major MCE SSJS and AMPScript function to its Salesforce Flow, Apex, and Data Cloud equivalent.

The hardest part of migrating to Marketing Cloud on Core is not the data. It is the code.

If you try to lift-and-shift your legacy ExactTarget scripts directly onto Core, your instance will crash. Marketing Cloud Engagement (MCE) let you run heavy SSJS and AMPScript exactly at the time of send. Core does not. It enforces strict platform governor limits to protect CRM processing power — and those limits will break every pattern you built over the past decade on the legacy platform.

At Genetrix, we have guided enterprise brands through this transition. To make the developer side of it tractable, we built an internal cheat sheet that maps every major legacy function directly to its native Salesforce Core equivalent. Today we are making it public.

Why Lift-and-Shift Fails on Marketing Cloud Core

The architectural difference is fundamental. In MCE, your automation logic lives inside the marketing layer — AMPScript runs at render time, SSJS runs inside Cloud Pages and Script Activities, and the CRM is essentially a data source you poll. In Core, the CRM is the platform. Your marketing logic must be a citizen of the Salesforce object model, which means it is subject to SOQL governor limits, Apex CPU limits, and Flow transaction boundaries.

> **The three most common breaking changes developers hit first:** LookupRows() at send-time exceeds SOQL governor limits. InsertData() fails against CRM field validation rules. HTTP.Post() with hardcoded tokens is a security violation under Core’s permission model.

You cannot just rebuild bad ExactTarget code 1:1 in Apex. You have to shift your logic “up” the funnel — into Flow, Data Cloud Transforms, and Invocable Methods that run before the send, not during it.

The Developer’s Translation Matrix: MCE to Core

The matrix is a 10-point technical cheat sheet mapping the SSJS and AMPScript functions your team uses every day to their direct equivalents in Salesforce Flow, Apex, and Data Cloud. Each mapping includes the reason the MCE approach fails on Core and the architectural principle behind the Core equivalent.

MCE Function	Core Equivalent
LookupRows()	Flow Get Records / Apex SOQL
InsertDE() / UpsertDE()	Apex DML / Flow Create / Update Records
HTTP.Post() at send-time	Named Credentials + Apex Callout (async)
WSProxy Execute()	Invocable Apex Method

Platform.Function.AutomationControl	Scheduled Flow / Batch Apex

The Full Matrix Covers 5 Technical Categories

- **CRUD & Database Operations** — LookupRows, InsertDE, UpsertDE, DeleteDE and their Core equivalents
- **Advanced Automation & Orchestration** — Automation Studio patterns mapped to Scheduled Flow and Batch Apex
- **Payload Parsing & APIs** — HTTP.Post/Get with hardcoded tokens vs Named Credentials and Apex Callouts
- **Cryptography & Security** — AMPScript encryption functions vs Platform Encryption and Shield
- **Array Manipulation & Transaction Control** — SSJS array handling vs Apex Collections and Flow loops

> **Download the MCE to Core Translation Matrix »**
>
> Free PDF · 10-point developer cheat sheet · High-res dark-mode format

How to Use This Matrix on a Live Migration

The matrix is most useful at the architecture review stage — before your developers write a single line of Core code. Walk through every custom AMPScript and SSJS function in your current MCE implementation, identify its Core equivalent in the matrix, and map out which ones require an architecture change (not just a syntax change) before migration begins.

Functions that require an architecture change — not just a rewrite — are the ones that will consume the most migration time. LookupRows() at send-time, for example, is not just a different function call on Core. It requires a fundamental shift in where that data lookup happens in your campaign flow.

> **Migration tip from Genetrix:** Prioritise translating your Triggered Send logic first. Triggered Sends in MCE rely heavily on AMPScript at render time. Core's equivalent — transactional messaging via the API — requires all personalisation data to be passed in the API payload, not looked up at render. This is the most common architecture gap we see in migration projects.

Frequently Asked Questions

Does Marketing Cloud on Core completely replace AMPScript and SSJS?

Not immediately and not entirely. AMPScript still works for email personalisation on Core — it has access to the Contact object and merge fields. What changes is that AMPScript can no longer be used for data operations (lookups, inserts, upserts) at send time, because those operations hit CRM governor limits. SSJS Script Activities are no longer available at all on Core — all server-side logic moves to Apex and Flow.

How long does a typical MCE to Core migration take?

For a mid-size enterprise with 50–150 active journeys and moderate custom code, Genetrix typically scopes 4–8 months for a full migration with parallel running. The developer translation work (rewriting SSJS/AMPScript logic into Flow and Apex) is usually 30–40% of the total migration effort.

Can we run MCE and Core in parallel during migration?

Yes — this is the recommended approach. Salesforce supports a phased migration where your existing MCE org continues running while your Core implementation is built and tested. The connector between the two platforms is MC Connect, which synchronises contacts and engagement data bidirectionally during the transition period.

What is the biggest mistake teams make on MCE to Core migrations?

Treating it as a technical migration instead of an architectural redesign. Teams that assign it to a single developer to “port the code” consistently run into governor limit failures and performance issues on Core. The migration needs an architect who understands both the MCE platform deeply and the Salesforce CRM platform deeply — which is a rare combination in the market.

Planning a Marketing Cloud Core Migration? Let’s Talk.

Genetrix has guided enterprise brands through MCE to Core migrations from architecture review through go-live. If you are planning a migration or want an expert assessment of your current MCE implementation, our team is ready to engage.

Get in Touch with Genetrix »