FLAM
DEVELOPED BY LIMES

FLAM Schulung – Ihr Wunschprogramm

Fragebogen und Themenkatalog

Unsere Schulungen sind kostenfrei. Sie stellen Raum und Zeit, wir bringen Referenten und Material mit. Damit wir Ihnen ein Programm zusammenstellen, das zu Ihren Leuten und zu Ihrem Einsatz von FLAM passt, beantworten Sie bitte die Fragen auf dieser Seite und markieren Sie im Katalog dahinter, was Sie interessiert. Das Formular ist ausfüllbar – öffnen Sie es in einem PDF-Reader, kreuzen Sie an, und senden Sie es an education@flam.de zurück.

1. Ihre Schulung

Wer nimmt teil?

☐ Systemprogrammierung, Betrieb ☐ Anwendungsentwicklung ☐ Entscheider, Architektur
☐ Informationssicherheit, Revision

Wie viel FLAM kennen die Teilnehmer schon?

☐ Anfänger – FLAM ist neu ☐ Profis – FLAM ist im Einsatz ☐ Gemischt

Wofür setzen Sie FLAM ein?

☐ Datenaustausch ☐ Archivierung ☐ Beides

Auf welchen Plattformen?

☐ z/OS ☐ Linux, Unix ☐ Windows ☐ Cloud, Objektspeicher

Über welchen Weg rufen Sie FLAM auf?

☐ Utility im Batch oder Skript ☐ Subsystem, Dateisystem – ohne Änderung der Anwendung
☐ Programmierschnittstelle ☐ Noch offen – bitte beraten

Wie soll die Schulung ablaufen?

☐ Bei Ihnen vor Ort ☐ Remote ☐ Mit Übungen am eigenen System
☐ Mit Besprechung eigener Anwendungsfälle

Rahmen

Gewünschte Dauer in Tagen

Wunschtermin oder Zeitraum

Zahl der Teilnehmer

Sprache der Schulung

Kontakt

Firma

Name

E-Mail

Telefon

2. Themenkatalog

Je Einheit drei Möglichkeiten: Schwerpunkt heißt ausführlich mit Übungen, kurz heißt ein Überblick, weglassen heißt nicht behandeln. Wer eine ganze Gruppe gleich behandeln will, markiert die Zeile der Gruppe. Ohne Markierung stellen wir die Einheit nach den Antworten von Seite 1 zusammen. Die ausführliche Beschreibung jeder Einheit steht im Anhang.

	Schwerpunkt	kurz	weglassen
Ebene 0 – Grundlagen			
1. FLAM-Grundlagen Was FLAM ist, woraus es besteht und wie man es erreicht	☐	☐	☐
2. Werkzeuge: FLCL und FLCC Die Kommandozeile, das Framework darunter und die GUI darüber	☐	☐	☐
3. Doku und Support Dokumentation, Downloadbereich, Ticketweg und Newsletter	☐	☐	☐
4. Die Kommandosprache CLE/P Wie ein Parameterstring gebaut ist – und was die Sprache selbst kann	☐	☐	☐
5. Die eingebauten Funktionen Nachschlagen, vorbelegen, Umgebung setzen, diagnostizieren	☐	☐	☐
6. Dateien auswählen, Namen bilden Wildcards, Member, Directory-Walk – und wie der Ausgabename entsteht	☐	☐	☐
7. Administration Konfiguration, Umgebung, Bibliotheken, Lizenz, Berechtigungen	☐	☐	☐
8. Betrieb und Wartung Qualitätssicherung, Auslieferung, Installation, Update	☐	☐	☐
Ebene 1 – Zugangswege			
9. Die drei Zugangswege Batch, Subsystem und API im Vergleich – und was FLAM4 von FLAM5 unterscheidet	☐	☐	☐
10. Zugang: Batch-Utility flcl in bash, PowerShell und JCL – und wie man das Ergebnis auswertet	☐	☐	☐
11. Zugang: Subsystem und Filesystem Ein DD-Statement auf z/OS, ein Mount auf Linux und Unix	☐	☐	☐
12. Zugang: die APIs Byte, Satz, Element, Unterprogramm – plus C++ und Java	☐	☐	☐
13. Netzwerk: SSH und FLIES SSH statt Pfad – und was aus FLIES geworden ist	☐	☐	☐
Ebene 2 – Funktionsthemen			
Archiv	☐	☐	☐
14. FLAM4 - Architektur und Schnittstellen Vier Eingänge in ein Zugriffsverfahren – und wie man den richtigen wählt	☐	☐	☐

	Schwerpunkt	kurz	weglassen
15. Das FLAM5-Archiv Aufbau, Schlüssel, Versionierung und Ablage	☐	☐	☐
16. Indizierung, Filter und Suche Wie FLAM5 in verschlüsselten Beständen sucht – und warum das wenig kostet	☐	☐	☐
17. Verwaltung archivierter Daten Suchen, ändern, einfügen, löschen – und prüfen	☐	☐	☐
18. Topfdesign Kompression, Verschlüsselung und Signatur je Spalte festlegen	☐	☐	☐
19. Segment-Store und Cloud Wo ein FLAM5-Archiv liegt – lokal, remote, in der Cloud oder im eigenen Speicher	☐	☐	☐
Daten und Formate	☐	☐	☐
20. Konvertierung mit FLUC Lesen, konvertieren, mehrfach schreiben	☐	☐	☐
21. Kompression Sieben Verfahren, ein gemessener Vergleich	☐	☐	☐
22. Dateiformate und Sätze Satzformate, Längenfelder, Trennzeichen – und die automatische Erkennung	☐	☐	☐
23. Tabellen und CSV Formate erkennen, Spalten umbauen, typgerecht ausgeben	☐	☐	☐
24. Der Format-Compiler Aus der Strukturbeschreibung wird die Tabellenbeschreibung	☐	☐	☐
25. Daten erkennen ohne Beschreibung Aus unbeschriebenen Sätzen wird eine Tabelle	☐	☐	☐
26. Zeichensätze 186 Codepages, vier Schlüsselfragen – und was auf dem Host anders ist	☐	☐	☐
27. XML-Verarbeitung XML lesen, schreiben – und in Zeilen und Spalten wandeln	☐	☐	☐
28. Fremdformate und Austausch ZIP, Base64, FLAMFILE – und das transparente Lesen	☐	☐	☐
Sicherheit	☐	☐	☐
29. Krypto und Schlüssel Verschlüsseln, signieren, Schlüssel verwalten	☐	☐	☐
30. PGP und Signaturen OpenPGP nach Norm, Hashes und Signaturen	☐	☐	☐
31. Sicherheit im Betrieb Was der Betreiber beachten muss – und welche Hintertüren absichtlich offen sind	☐	☐	☐
32. SAF und RACF Funktionen eines Programms über RACF kontrollieren	☐	☐	☐

	Schwerpunkt	kurz	weglassen
Werkzeuge	☐	☐	☐
33. Hilfsfunktionen Auskunft, Suche, Prüfsummen, Vergleich, Schlüssel	☐	☐	☐
Betrieb	☐	☐	☐
34. Einbettung in Batch-Jobs Best Practices für JCL, Parameterdateien und Returncodes	☐	☐	☐
35. Performance und Optimierung Warum FLAM für Massendaten gebaut ist – und was Sie einstellen können	☐	☐	☐
36. Beschleuniger auf z/OS zEDC, CPACF und zIIP – was wirklich hilft und was nicht	☐	☐	☐

3. Die Einheiten im Einzelnen

Zu jeder Einheit des Katalogs die ausführliche Beschreibung, in derselben Reihenfolge und mit derselben Nummer.

1. FLAM-Grundlagen

Was FLAM ist, woraus es besteht und wie man es erreicht

Die Einheit, mit der jede Schulung beginnt: FLAM ist ein Zugriffsverfahren und kein Komprimierungswerkzeug – es liest und schreibt Daten, und Kompression, Verschlüsselung, Signatur und Zeichenwandlung passieren dabei. Sie lernen die drei Komponentengruppen FLC/ORG, FLM und FLS auseinanderzuhalten und damit zu sagen, welcher Teil überhaupt klare Daten sieht. Dazu der Satz, der die ganze Schulung trägt: ein Parameterstring, drei Wege hinein. Am Ende können Sie einordnen, wo Sie für Ihre Frage nachschlagen, und kennen die Plattform- und Release-Politik.

2. Werkzeuge: FLCL und FLCC

Die Kommandozeile, das Framework darunter und die GUI darüber

Die Werkzeugeinheit: 15 Kommandos in einem Programm, dazu 25 eingebaute Funktionen und eine GUI, die den Aufrufstring zusammenbaut. Sie lernen die Syntax der Kommandosprache zu lesen – Kommando, Objekt, Overlay, Array, Zahl, Zeichenkette –, die eingebaute Hilfe zu benutzen und sich Teile des Handbuchs selbst zu erzeugen. FLCC ist dabei mehr als eine Oberfläche: es ist das interaktive Handbuch und der Generator für Shell-Zeile, Parameterdatei und JCL. Wer hier mitgeht, tippt danach keinen langen String mehr von Hand.

3. Doku und Support

Dokumentation, Downloadbereich, Ticketweg und Newsletter

Wo steht was, und wie kommt man an Hilfe. Ein Handbuch gilt gemeinsam für Utility, Subsystem und Schnittstelle, weil alle drei denselben Parameterstring nehmen – das erklärt seinen Aufbau. Sie lernen, auf flam.de zielsicher Dokumentation, Software, Addons und Bibliotheken zu finden, und ein Ticket so zu stellen, dass es ohne Rückfrage bearbeitet werden kann. Dazu die Regel, die einen Datenschutzvorfall verhindert: keine Dumps und keine produktiven Daten an den Support.

4. Die Kommandosprache CLE/P

Wie ein Parameterstring gebaut ist – und was die Sprache selbst kann

Die Sprache, in der jeder FLAM-Aufruf geschrieben ist – und zwar auf allen Wegen gleich. Sie lernen die fünf Parameterformen, die Lexeme mit ihren Kommentaren und Zahlenformaten und die Operatoren, mit denen ein String kürzer wird: der Wert aus einer Datei für Passphrasen, die Parameterdatei je Kommando, Overlay und Objekt. Am Ende lesen Sie eine Syntaxzeile aus dem Handbuch ohne Hilfe und wissen, warum ein Aufruf mit einer Klammer mehr etwas anderes bedeutet.

5. Die eingebauten Funktionen

Nachschlagen, vorbelegen, Umgebung setzen, diagnostizieren

Die 25 eingebauten Funktionen sind kein Beiwerk, sondern der Rahmen um die Kommandos: Syntax und Hilfe, Handbuchseiten und Dokumentation erzeugen, Properties vorbelegen und pflegen, Umgebungsvariablen setzen und anzeigen, Version, Lizenz und Fehlerliste ausgeben. Sie lernen, welche Funktion in welcher Situation hilft, wie eine Property-Datei entsteht und wie man mit dem Owner mehrere Ausprägungen nebeneinander betreibt – Test gegen Produktion, oder einen Mandanten je Kunde.

6. Dateien auswählen, Namen bilden

Wildcards, Member, Directory-Walk – und wie der Ausgabename entsteht

Ein Aufruf, viele Dateien: diese Einheit zeigt, wie man Dateien auswählt und wie aus jedem Eingabenamen der passende Ausgabename entsteht. Sie lernen die Wildcards und ihre Grenzen, die Auswahl von Membern in Bibliotheken und Archiven, den Directory-Walk über Unterverzeichnisse und die Namensbildung mit Platzhaltern. Dazu die Namensregeln der Plattformen: wann ein Name ein Pfad ist, wann ein Dataset und wann eine Zuweisung aus der JCL – und was passiert, wenn eine Umgebungsvariable nicht definiert ist.

7. Administration

Konfiguration, Umgebung, Bibliotheken, Lizenz, Berechtigungen

Die Einheit für die, die FLAM betreiben. Fünf Quellen bestimmen, was ein Aufruf tut – Kommandozeile, Parameterdatei, Property-Datei, Konfiguration und Umgebung –, und ihre Rangfolge im Kopf zu haben erspart die häufigste Supportfrage: warum wirkt mein Parameter nicht. Sie lernen sich alle Umgebungsvariablen mit ihrer Bedeutung anzeigen zu lassen, welche Bibliotheken und Lademodule Ihre Plattform braucht, wie die Lizenz eingebunden wird und wo die Berechtigungen ansetzen.

8. Betrieb und Wartung

Qualitätssicherung, Auslieferung, Installation, Update

Was in einer Auslieferung steckt und wie man sie in den Betrieb bringt. Sie erfahren, mit welchen Verfahren wir vor der Auslieferung prüfen, was signiert ist und woran man es nachrechnet, und wie ein Update ohne Stillstand läuft. Dazu die Antwort auf die Frage, die in jedem Audit kommt: wie beantworte ich eine CVE-Frage zu FLAM und zu den externen Bibliotheken darin – mit den beiden Listen, die wir zu jeder Version veröffentlichen.

9. Die drei Zugangswege

Batch, Subsystem und API im Vergleich – und was FLAM4 von FLAM5 unterscheidet

Die Einheit, die vor der Entscheidung steht: es gibt drei Wege in FLAM hinein, und alle drei nehmen denselben Parameterstring. Sie lernen, für Ihren Fall den passenden Weg zu begründen – und zwar nach der Frage, wer etwas ändern muss: der Betrieb oder die Entwicklung. Dazu die Abgrenzung der Generationen: was FLAM4 kann, was FLAM5 dazu kann und warum beides nebeneinander betrieben wird. Am Ende erkennen Sie denselben Aufruf in jeder Verpackung wieder.

10. Zugang: Batch-Utility

flcl in bash, PowerShell und JCL – und wie man das Ergebnis auswertet

Der Weg, den die meisten Häuser zuerst gehen: FLAM als Kommando in einer Jobkette. Sie lernen, denselben Aufruf in bash, PowerShell und JCL zu schreiben, lange Parameter in Dateien auszulagern, und den Completion Code richtig zu deuten – inklusive der Klammer, mit der aus einer Warnung keine rote Kette wird. Dazu die Allokation beim Schreiben: warum eine dynamische Allokation die bessere Wahl ist und was eine statische Zuweisung mit der Form Ihrer Sätze macht.

11. Zugang: Subsystem und Filesystem

Ein DD-Statement auf z/OS, ein Mount auf Linux und Unix

Der transparente Weg: die Anwendung bleibt unverändert, und FLAM hängt sich unter sie. Auf z/OS über ein DD-Statement mit dem Subsystem, auf Linux und Unix über ein eingehängtes Verzeichnis. Sie lernen, wie der Parameterstring in dieser Welt übergeben wird, was das Subsystem an Konvertierung, Kompression und Verschlüsselung leisten kann und was auf diesem Weg ausdrücklich nicht geht. Der Fall, der überzeugt: ein unveränderter Utility schreibt eine komprimierte Datei auf einen fernen Rechner.

12. Zugang: die APIs

Byte, Satz, Element, Unterprogramm – plus C++ und Java

Die Einheit für die Anwendungsentwicklung: vier Schnittstellen, die sich in der Abstraktionsebene unterscheiden, nicht im Funktionsumfang. Sie lernen, welche Schnittstelle zu welcher Aufgabe passt – Bytes, Sätze, Elemente oder der Aufruf eines Unterprogramms mit demselben Parameterstring, den die Kommandozeile nimmt. Dazu die Sprachbindungen für C++ und Java und die Regeln, die man beim Einbinden in eine bestehende Anwendung kennen muss.

13. Netzwerk: SSH und FLIES

SSH statt Pfad – und was aus FLIES geworden ist

Lesen und schreiben über das Netz, ohne einen eigenen Server zu betreiben: eine URL steht dort, wo sonst der Pfad steht. Sie lernen die zwei Schreibweisen für dieselbe Verbindung, die fünf Stufen der Hostkey-Prüfung und die batchtaugliche darunter, die Nutzung der vorhandenen OpenSSH-Konfiguration und die Vor- und Nachverarbeitung auf der Gegenseite. Dazu die klare Auskunft, was aus den alten FLIES-Verfahren geworden ist und was heute an ihrer Stelle steht.

14. FLAM4 - Architektur und Schnittstellen

Vier Eingänge in ein Zugriffsverfahren – und wie man den richtigen wählt

FLAM4 ist kein Komprimierungswerkzeug mit angebautem API, sondern ein Zugriffsverfahren mit vier Eingängen: Utility, Subsystem, Satzschnittstelle und Unterprogramm. Sie lernen, diese vier auseinanderzuhalten und für eine gegebene Datei den passenden zu wählen, und Sie sehen, wie ein einzelner Satz in einem großen Archiv gefunden wird. Dazu FKME: wie die Schlüsselverwaltung im Haus bleibt. Die Einheit ordnet außerdem ein, welche Grenzen FLAM4 hat und wo FLAM5 anfängt.

15. Das FLAM5-Archiv

Aufbau, Schlüssel, Versionierung und Ablage

Das FLAM5-Archiv von innen: Matrizen, Segmente und die drei Schlüssel, mit denen Verzeichnis, Daten und Index getrennt geschützt werden. Sie lernen, wie ein Archiv wächst, was eine Version ist und warum ein Betreiber ein Archiv verwalten kann, ohne die Daten zu sehen. Dazu die Trennung von Archivverwaltung und Ablage, die erklärt, warum dasselbe Archiv lokal, auf einem fernen Rechner oder in der Cloud liegen kann.

16. Indizierung, Filter und Suche

Wie FLAM5 in verschlüsselten Beständen sucht – und warum das wenig kostet

Die Einheit, die erklärt, wie man in komprimierten, verschlüsselten und signierten Beständen sucht, ohne sie klarzurechnen. Sie lernen die Mechanik des Bloom-Filters vom Hash über die Teilsignatur zur Segmentsignatur, entwerfen eine Indizierung je Spalte – wie viele Hashes, wie lang die Signatur – und verstehen den zweistufigen Zugriff, bei dem ein Treffer im Filter noch kein Treffer im Satz ist. Dazu die Grenzen und die Frage, wann eine echte Datenindizierung die bessere Wahl wäre.

17. Verwaltung archivierter Daten

Suchen, ändern, einfügen, löschen – und prüfen

Was man mit einem bestehenden Archiv tut: suchen, Sätze ändern, einfügen und löschen, Versionen verwalten und aufräumen – alles ohne die ganze Datei klarzurechnen. Sie lernen die Kommandos dafür, die Ergebnismengen und ihre Weiterverarbeitung, die Prüftiefen beim Verifizieren und den Unterschied zwischen Löschen und Aufräumen. Dazu die Kapazitätsfrage: wie ein Archiv mit vielen Änderungen wächst und wann es sich lohnt, es neu zu schreiben.

18. Topfdesign

Kompression, Verschlüsselung und Signatur je Spalte festlegen

Das Topfdesign ist Entwurfsarbeit, keine Konfiguration: ein Topf ist eine Gruppe von Spalten, für die gemeinsam gilt, wie komprimiert, verschlüsselt und signiert wird. Sie lernen die zwei Fragen, mit denen man Spalten gruppiert, die Parameter eines Topfes mit ihren drei Strömen und die Abwägung dahinter: ein Topf stört die Kompression, zu viele Töpfe haben je Topf zu wenig Daten. Dazu die Vorbelegungen des Archivs, unter ihnen die Integritätsstufe – die Antwort auf die Sicherheitsfrage.

19. Segment-Store und Cloud

Wo ein FLAM5-Archiv liegt – lokal, remote, in der Cloud oder im eigenen Speicher

Die untere Schicht des Archivs: wo die Segmente physisch liegen. Sie lernen die Ablagen kennen – Datei, Ordner, S3, Azure, IBM CDA und der Zugriff über SSH –, die Pufferung, ohne die ein Archiv in der Cloud unbenutzbar langsam wäre, und die beiden Programmierschnittstellen, mit denen ein Haus einen eigenen Speicher anbindet. Dazu die ehrliche Trennung zwischen dem, was heute geht, und dem, was als cloudFLAM geplant ist.

20. Konvertierung mit FLUC

Lesen, konvertieren, mehrfach schreiben

Der Kern des Konverters: ein Eingang, beliebig viele Konvertierungen, mehrere Ausgänge – und alles in einem Schritt, ohne temporäre Dateien. Sie lernen, welches Leseobjekt zu welcher Quelle passt und welche Kombinationen aus Lesen und Schreiben sinnvoll sind, wie Zeichensätze auf drei Ebenen angegeben werden und wie aus einer Kette von neun Jobschritten ein einziger wird. Der Fall aus der Praxis wird dabei vollständig durchgerechnet.

21. Kompression

Sieben Verfahren, ein gemessener Vergleich

Welches Kompressionsverfahren wann, und was es kostet. Sie sehen einen Vergleich der sieben unterstützten Verfahren mit gemessenen Zahlen, lernen warum beim Lesen keine Angabe nötig ist, und was Kompression mit den Satzgrenzen macht. Dazu die Einordnung der Hardwarebeschleunigung auf dem Mainframe und der Fall, in dem man ein Verfahren nicht wegen seiner Rate nimmt, sondern weil die Gegenseite es erwartet.

22. Dateiformate und Sätze

Satzformate, Längenfelder, Trennzeichen – und die automatische Erkennung

Die Einheit über die Form der Daten: feste und variable Sätze, Längenfelder und Trennzeichen, Druckaufbereitungszeichen und die Formate der Mainframe-Welt. Sie lernen, ein Format anzugeben, es aus dem Dateinamen lesen zu lassen oder es FLAM an den Daten erkennen zu lassen – und wo die Erkennung an ihre Grenze kommt. Dazu die Netto-Regel für die Satzlänge und die beiden Randfälle, die Supportfälle erzeugen: der leere und der sehr große Satz.

23. Tabellen und CSV

Formate erkennen, Spalten umbauen, typgerecht ausgeben

Aus Sätzen werden Zeilen und Spalten: diese Einheit zeigt, wie eine Struktur beschrieben wird, wie FLAM sie in vielen Fällen selbst erkennt und wie daraus CSV, JSON oder eine andere Tabellendarstellung wird. Sie lernen die Spaltentypen und ihre Formate kennen, den Umbau von Spalten und die drei Kommandoprodukte für Binär-, ASCII- und Fremdformat-CSV. Der rote Faden ist die Frage, wer die Beschreibung hat – Sie, ein Copybook oder niemand.

24. Der Format-Compiler

Aus der Strukturbeschreibung wird die Tabellenbeschreibung

Wer eine Struktur beschrieben hat, muss sie nicht noch einmal beschreiben: FLAM übersetzt ein COBOL-Copybook, eine PL/I-Strukturdefinition oder eine HLASM-DSECT selbst in die Tabellenbeschreibung. Diese Einheit zeigt, was der Compiler aus den Feldern macht, was er mitbringt, das in der Quelle gar nicht steht, wie vier Anmerkungen am Feld ihn steuern und wie IMS-Entladungen Segment für Segment ihr Copybook bekommen.

25. Daten erkennen ohne Beschreibung

Aus unbeschriebenen Sätzen wird eine Tabelle

Wenn es kein Copybook mehr gibt, bleibt die Datei trotzdem lesbar: FLAM leitet die Spaltenstruktur aus den Daten selbst ab. Diese Einheit zeigt die drei Erkenner für feste Sätze, zeichengetrennte Daten und XML, woran ein Erkenner eine Spaltengrenze festmacht, wo er sie nicht finden kann, wie Hinweise den Suchraum verengen und wie aus einer IMS-Entladung ohne Segmentbeschreibung eine CSV-Datei wird. Und sie zeigt den Weg zurück: eine erkannte Beschreibung ansehen, korrigieren und festschreiben.

26. Zeichensätze

186 Codepages, vier Schlüsselfragen – und was auf dem Host anders ist

Zeichensätze sind die häufigste Ursache für Daten, die falsch aussehen. Sie lernen, die vier Fragen zu trennen, die dabei immer zusammengeworfen werden: in welchem Zeichensatz liegen die Daten, in welchem sollen sie hinterher liegen, was passiert mit Zeichen, die es im Ziel nicht gibt, und in welchem Zeichensatz ist eigentlich der Aufruf getippt. Dazu die Eigenheiten der EBCDIC-Welt, die Unicode-Formen mit ihren Kennungen und die Werkzeuge, mit denen man eine Wandlung nachprüft.

27. XML-Verarbeitung

XML lesen, schreiben – und in Zeilen und Spalten wandeln

XML ist für FLAM ein Datenformat wie andere: es wird gelesen, konvertiert und geschrieben. Sie lernen, wie ein Dokument in Elemente zerlegt wird, wie aus XML eine Tabelle und aus einer Tabelle XML wird, was mit Namensräumen, CDATA und Zeilenenden passiert, und wie man ein Dokument minimiert oder einrückt. Dazu die Grenzen, die man kennen sollte, bevor man ein fremdes Format an den Parser gibt.

28. Fremdformate und Austausch

ZIP, Base64, FLAMFILE – und das transparente Lesen

Der Austausch mit Systemen, die FLAM nicht haben: ZIP-Archive nach Norm, Base64 und andere Kodierungen, GZIP und die alten FLAMFILEs. Sie lernen, wie FLAM beim Lesen selbst erkennt, was es vor sich hat, wie man ein Archivmember gezielt anspricht und welche Kodierung zu welchem Zielsystem passt. Dazu die Regel, die Zeit spart: das Ziel einer Kodierung ist ein Binärstrom, nicht ein Text.

29. Krypto und Schlüssel

Verschlüsseln, signieren, Schlüssel verwalten

Die Einheit, in der aus Verfahren eine Betriebsentscheidung wird. Sie lernen die Verfahren kennen, die FLAM anbietet, den Unterschied zwischen Passphrase und Schlüsselverwaltung, und wie FKME die Schlüssel im Haus hält. Dazu die Fragen, die vor der Technik kommen: wer darf entschlüsseln, wer verwaltet die Schlüssel, was passiert beim Personalwechsel – und wie man die Antworten in Parameter übersetzt.

30. PGP und Signaturen

OpenPGP nach Norm, Hashes und Signaturen

PGP ist der gemeinsame Nenner im Datenaustausch zwischen Häusern. Sie lernen, wie FLAM OpenPGP nach RFC umsetzt, wie Schlüssel importiert und verwaltet werden, was ASCII-Armor ist und wann man ihn braucht. Dazu die Hash- und Signaturverfahren mit ihren Einsatzfällen und der Unterschied, auf den es im Audit ankommt: eine Prüfsumme belegt Unverändertheit, eine Signatur belegt zusätzlich, wer es war.

31. Sicherheit im Betrieb

Was der Betreiber beachten muss – und welche Hintertüren absichtlich offen sind

Die Einheit für die Sicherheitsabteilung: was FLAM im Betrieb voraussetzt, wo die klaren Daten unterwegs sind und welche Möglichkeiten es gibt, die ein Angreifer benutzen könnte. Sie lernen die Schalter kennen, mit denen ein Haus sich abschottet, und die Stellen, die absichtlich offen sind, weil ein Betrieb sie braucht. Der Ton ist offen: wir nennen die Grenzen selbst, statt sie auf Nachfrage zuzugeben.

32. SAF und RACF

Funktionen eines Programms über RACF kontrollieren

Auf z/OS lässt sich nicht nur der Zugriff auf Dateien regeln, sondern die Nutzung der Funktionen von FLAM selbst – über eine eigene SAF-Klasse. Sie lernen, wie eine Ressource aufgebaut ist, was kontrollierbar ist (Richtlinien, Parameterstrings, Schnittstellen, URLs, Verbindungen, Schlüssel, Protokoll und Lizenz), wie man die Klasse anlegt und aktiviert und in welcher Reihenfolge man einführt, ohne den Betrieb anzuhalten. Dazu die ehrliche Auskunft, was auf anderen Plattformen davon fehlt.

33. Hilfsfunktionen

Auskunft, Suche, Prüfsummen, Vergleich, Schlüssel

Die kleinen Kommandos, die im Alltag die meiste Zeit sparen: Auskunft über System und Dateien, Suche mit regulären Ausdrücken auch auf dem Host, Prüfsummen, der logische Vergleich zweier Dateien und die Prüfung gegen eine Formatbeschreibung. Sie lernen, welches Kommando welche Frage beantwortet – Bytes, Inhalt oder Struktur – und welches davon in ein Ticket gehört, bevor Sie eines aufmachen.

34. Einbettung in Batch-Jobs

Best Practices für JCL, Parameterdateien und Returncodes

Die Einheit, die aus einer Jobkette einen Schritt macht. Sie sehen, wie aus sieben temporären Dateien ein Aufruf wird, wie Parameterdateien und JCL-Symbole eine Prozedur wiederverwendbar machen, wie Returncodes gedeutet werden und was in der Umgebung des Steps stehen sollte. Dazu die Schalter für Aufräumen, Umbenennen und leere Ergebnisse – die Punkte, an denen eine Automatisierung sonst stehen bleibt.

35. Performance und Optimierung

Warum FLAM für Massendaten gebaut ist – und was Sie einstellen können

Die Einheit mit den Zahlen: was der Durchsatz kostet, wo die Zeit hingeht und welche Stellschrauben wirklich wirken. Sie lernen die Wirkung von Blockgröße und Satzanzahl, den Unterschied zwischen einem Lauf über viele Dateien und vielen Läufen, die Kosten der Zeichenwandlung und die Protokollierung als oft vergessenen Posten. Gerechnet wird mit gemessenen Werten, nicht mit Schätzungen.

36. Beschleuniger auf z/OS

zEDC, CPACF und zIIP – was wirklich hilft und was nicht

Die Einheit für den Mainframe-Betrieb: welche Hardwarebeschleuniger FLAM nutzt, wie man das nachprüft und was sie tatsächlich bringen. Sie lernen die Verfahren auseinanderzuhalten – Kompression in der Karte, Kryptobefehle im Prozessor, verlagerungsfähige Arbeit – und die ehrliche Antwort auf die Frage nach dem Offloading. Dazu der Betrieb der Started Task und die Stellen, an denen ein Update mehr verlangt als das Austauschen einer Bibliothek.

4. Bemerkungen

Was hier nicht abgefragt wurde, gehört hierher: eigene Anwendungsfälle, die Sie besprechen möchten, Dateien oder Formate, die wir uns vorher ansehen sollen, organisatorische Randbedingungen.

Unser Ticket / Angebot (von uns ausgefüllt)

Senden Sie das ausgefüllte Formular an education@flam.de. Wir stellen daraus ein Programm zusammen, schlagen Ihnen Termin und Dauer vor und schicken Ihnen die Unterlagen zu den gewählten Einheiten vorab als PDF.