



PROGRAMME

Matin (9h – 12h30)

- Introduction et prérequis
- Débogage simple
 - Construction d'un Job de démonstration
 - Utilisation du mode Traces
 - Configuration des colonnes affichées
 - Utilisation des points d'arrêt simples
 - Analyse de l'exécution en mode Debug
- Débogage avancé
 - Identification des anomalies fonctionnelles et des erreurs techniques
 - Utilisation de tLogRow pour observer les données
 - Gestion des flux valides et des rejets avec tMap
 - Gestion d'une erreur technique avec OnSubjobError et tWarn
 - Présentation de tLogCatcher
 - Application pratique sur un flux clients
- Exécution parallèle
 - Différence entre exécution séquentielle et multi-thread
 - Activation de l'option Exécution multi-thread
 - Identification des sous-jobs pouvant être exécutés simultanément
 - Comparaison des temps d'exécution avec le Job BigFiles
 - Limites et précautions liées au multi-thread
 - Écriture parallèle en base de données (module optionnel selon l'environnement disponible)
- Exercices pratiques

À l'issue de la demi-journée, le stagiaire sera capable de diagnostiquer un Job Talaxie, de gérer différents types d'anomalies et de mettre en oeuvre une exécution multi-thread sur des sous-jobs indépendants.

Après-midi (14h – 17h)

- Mémorisation des données
 - Présentation du composant tMemorizeRows
 - Importance de l'ordre des données
 - Utilisation de tSortRow et tMemorizeRows
 - Comparaison entre une ligne courante et une ligne précédemment traitée
 - Application pratique : récupération du prix précédent d'un produit
- Exercices pratiques
- Utilisation de code et de routines Java
 - Différences entre tJava, tJavaRow et tJavaFlex
 - Choix du composant selon le besoin
 - Transformation d'un flux avec tJavaRow
 - Présentation de tJavaFlex et de ses phases Start / Main / End
 - Création et utilisation d'une routine Java réutilisable
- Évaluation théorique des acquis
- Exercices pratiques

Après-midi (17h – 17h30)

- Evaluation à chaud

À l'issue de la demi-journée, le stagiaire sera capable de mémoriser des valeurs entre plusieurs lignes, d'utiliser les principaux composants Java de Talaxie et de créer une routine Java réutilisable.

Objectifs

Permettre aux utilisateurs de maîtriser les fonctionnalités avancées de **Talaxie Data Integration** afin de mieux diagnostiquer les traitements, optimiser l'exécution des Jobs et mettre en oeuvre des transformations avancées. La formation aborde notamment le débogage et la gestion des erreurs, l'exécution multi-thread de sous-jobs indépendants, la mémorisation de données avec **tMemorizeRows**, ainsi que l'utilisation de composants et de routines Java réutilisables.

Public concerné

- Utilisateurs de **Talaxie Data Integration**
- Développeurs, intégrateurs, consultants ou responsables de développement disposant déjà des connaissances de base sur Talaxie.

Pré-requis

- Talaxie installé et fonctionnel sur chaque poste stagiaire
- Avoir suivi la formation Talaxie Data Integration Basics ou disposer de connaissances équivalentes
- Disposer de notions de base en Java pour la partie consacrée aux composants Java et aux routines
- Disposer des fichiers et ressources nécessaires aux exercices pratiques
- Un accès à une base de données est nécessaire uniquement si le module optionnel d'écriture parallèle en base est réalisé.

Moyens et méthodes pédagogiques

- La formation sera assurée par un consultant formateur garant du respect du contenu pédagogique et de la méthodologie définis par DeciVision et les modèles Editeurs.
- Moyens : support pédagogique (PDF numérique remis à chaque stagiaire),
- Méthodes : Exposé, démonstration par le formateur, exercices pratiques pour les stagiaires