# Manuel de Référence - Usine à Tests V3.0

*Dernière mise à jour : 22 août 2025*

## Partie 1 : Vision, Principes et Architecture

### 1.1. Introduction et Vision du Projet

L'**Usine à Tests** est un système intégré et automatisé conçu pour gérer l'ensemble du cycle de vie des questionnaires au sein de l'écosystème Google Workspace.

**L'objectif global** est de transformer une simple ligne de configuration en un système de test complet et autonome, incluant :

* Un **Google Form** multilingue et dynamique.
* Une **Google Sheet** de traitement des réponses ("Kit de Traitement") dotée d'une logique de calcul et de communication.
* Un système d'envoi d'**e-mails de résultats** personnalisés, multilingues, et pouvant être différés.

Le système vise à réduire drastiquement le temps de déploiement, à éliminer les erreurs manuelles et à offrir une flexibilité maximale pour créer une large gamme de tests.

### 1.2. Contexte et Évolution

Le projet a maturé à travers plusieurs phases majeures pour atteindre sa version actuelle, stable et robuste :

* **Version 0 - Le Prototype Artisanal** : Un système "tout-en-un" où une seule feuille de calcul gérait un unique test de A à Z. Rapide mais peu scalable.
* **Version 1 - L'Usine Spécialisée** : Introduction de la séparation entre l'usine de génération [MOTEUR] et le produit [TEMPLATE]. Chaque "Kit" généré contenait cependant une logique de calcul qui lui était propre, rendant la maintenance complexe.
* **Version 2 - L'Usine Universelle (Actuelle)** : L'architecture actuelle où toute la logique de calcul et de communication est externalisée dans les données. Le code n'est plus qu'un interpréteur. Cette version a été considérablement renforcée par :
  + La mise en place d'un **moteur de calcul universel** piloté par le JSON.
  + La transformation du système d'e-mails en un **moteur d'assemblage dynamique** piloté par la base de données.
  + L'ajout de **fonctionnalités métier avancées** comme l'e-mail de confirmation et l'envoi différé.
  + Le développement d'un **framework de débogage** (LoggerLib) et d'interfaces utilisateur pour la configuration et le retraitement.

### 1.3. Principes Directeurs

Le développement du projet respecte trois principes fondamentaux qui garantissent sa robustesse et sa maintenabilité.

1. **Zéro Convention Implicite** : Tout paramètre ou toute logique doit être explicitement défini dans la configuration. Le code ne doit rien deviner.
   * *Exemple concret* : La résolution du bug des e-mails non reçus était due à une clé de traduction (PREFIXE\_COPIE\_EMAIL) manquante dans la BDD. Le système ne l'a pas inventée, prouvant la validité de ce principe.
2. **Le JSON comme Source de Vérité** : Toute la logique de calcul d'une question (type, profils, scores) est contenue dans le champ Paramètres (JSON) de la base de données. Le code n'est qu'un interpréteur.
   * *Exemple concret* : La logique du mode ECHELLE\_NOTE (bornes, labels) est entièrement définie dans le JSON, le script ne fait que lire ces paramètres pour construire la question.
3. **Le Code plus Intelligent que la Configuration** : Le code doit être conçu pour se défendre contre des erreurs de configuration et gérer des cas variés de manière robuste.
   * *Exemple concret* : L'ajout systématique de la fonction .trim() lors de la lecture des ID de pièces jointes a résolu un bug critique causé par un caractère de saut de ligne invisible, rendant le code insensible à ce type d'erreur de saisie.

### 1.4. Architecture Globale des Composants

Le système s'articule autour de cinq composants principaux qui interagissent en permanence.

* **1. [CONFIG] Usine à Tests (Google Sheet)**
  + **Rôle** : Le panneau de contrôle central. C'est ici que l'administrateur décrit les tests à générer via une interface de configuration (FormulaireUI.html) ou en modifiant directement l'onglet Paramètres Généraux.
  + **Scripts associés** : 02\_configuration\_Script\_250822.txt
* **2. [MOTEUR] Usine à Tests (Apps Script)**
  + **Rôle** : L'orchestrateur de construction. Ce script lit une ligne de configuration dans [CONFIG] et construit l'ensemble des fichiers (le Formulaire et le Kit de traitement).
  + **Scripts associés** : 01\_Moteur\_Script\_250822.txt
* **3. [BDD] Tests & Profils (Google Sheet)**
  + **Rôle** : La base de données centrale, le "cerveau" de l'usine. Elle contient toutes les données nécessaires au fonctionnement des tests.
  + **Scripts associés** : 03\_BaseDeDonnées\_Script\_250822.txt (utilitaires)
* **4. [TEMPLATE] Kit de Traitement V2 (Google Sheet)**
  + **Rôle** : Le modèle de feuille de calcul autonome qui est copié pour chaque nouveau test. Son script interne gère le traitement des réponses, le calcul des scores et la communication par e-mail.
  + **Scripts associés** : 04\_Templates\_Script\_250822.txt
* **5. LoggerLib (Apps Script Library)**
  + **Rôle** : Une librairie de script partagée et externe dont l'unique rôle est de fournir des fonctions de journalisation pour le débogage.

### 1.5. Dépendances et Prérequis

* **Librairie Externe** : Le [TEMPLATE] dépend de la librairie LoggerLib pour le débogage. Son ID de script doit être ajouté dans les dépendances du projet Apps Script du template.
* **Permissions Google Workspace** : Pour fonctionner, les scripts requièrent des autorisations de la part de l'utilisateur qui les exécute pour accéder à :
  + **Google Drive** (pour créer et gérer les fichiers).
  + **Google Sheets** (pour lire la configuration et écrire les résultats).
  + **Google Forms** (pour créer et modifier les questionnaires).
  + **Gmail** (pour envoyer les e-mails de confirmation et de résultats).

## Partie 2 : Guides d'Utilisation

### 2.1. Guide de Démarrage Rapide : Déployer un Test en 3 Étapes

Le moyen le plus simple et le plus fiable de créer un nouveau test est d'utiliser le processus de déploiement centralisé.

* **Étape 1 : CONFIGURER**
  1. Ouvrez la feuille de calcul [CONFIG] Usine à Tests.
  2. Dans le menu, cliquez sur **🚀 Actions Usine -> Configurer un nouveau test...**.
  3. Remplissez le formulaire qui apparaît dans la barre latérale. Il permet de définir le titre, le type de test, les questions standards (blocs méta) et les options d'envoi d'e-mail.
  4. Cliquez sur "Créer la Configuration". Une nouvelle ligne est ajoutée dans l'onglet Paramètres Généraux avec le statut **"En construction"**. Notez le numéro de cette ligne.
* **Étape 2 : DÉPLOYER**
  1. Ouvrez le projet Apps Script [MOTEUR] Usine à Tests.
  2. Dans le menu, cliquez sur **🏭 Usine à Tests -> 🚀 Déployer un test de A à Z...**.
  3. Une boîte de dialogue vous demandera le numéro de la ligne à déployer. Entrez le numéro noté précédemment.
* **Étape 3 : ACTIVER**
  1. Une fois le processus terminé, une fenêtre de confirmation s'affiche avec deux liens. Cliquez sur le second : **"Ouvrir le Kit pour l'activer"**.
  2. Dans le menu de cette nouvelle feuille de calcul (le "Kit de Traitement"), cliquez une seule fois sur **⚙️ Actions Usine -> Activer le traitement automatique**.

**Votre test est maintenant 100% opérationnel.** Le lien public du formulaire est disponible dans la fenêtre de confirmation et dans la colonne Lien\_Formulaire\_Public de votre feuille [CONFIG].

### 2.2. Guide des Opérations Courantes

* Retraiter une Réponse  
  Pour relancer le calcul et l'envoi d'un e-mail pour une soumission existante :
  1. Ouvrez le "Kit de Traitement" concerné.
  2. Allez dans le menu **⚙️ Actions Usine -> Retraiter une réponse...**.
  3. Entrez le **numéro de la ligne** à retraiter.
  4. Une **barre latérale** s'ouvrira, vous permettant de forcer une langue, un niveau de résultat, un alias d'expédition ou de modifier les destinataires avant de lancer le retraitement.
* **Ajouter des Pièces Jointes PDF**
  1. Ouvrez la feuille de calcul [BDD] Tests & Profils.
  2. Dans le menu, cliquez sur **⚙️ Utilitaires BDD -> Lister les fichiers d'un dossier Drive**.
  3. Entrez l'ID du dossier Drive contenant vos fichiers PDF.
  4. Les fichiers seront listés dans l'onglet Liste\_Fichiers\_Drive.
  5. Vous pouvez maintenant référencer ces fichiers dans l'onglet sys\_Composition\_Emails pour qu'ils soient attachés aux e-mails.

## Partie 3 : Dictionnaire des Données et Configuration

### 3.1. Le Dictionnaire Complet des Données

#### Fichier [CONFIG]V2 Usine à Tests

C'est le panneau de contrôle de l'usine.

* **Onglet Paramètres Généraux** : Le "carnet de commandes".
  + Type\_Test : Code du test (ex: "Couleurs"). Détermine les questions et profils à utiliser.
  + Moteur\_Calcul : Doit être **"Universel"** pour la version actuelle.
  + Blocs\_Meta\_A\_Inclure : Liste des ID des questions standards (ex: META\_EMAIL,META\_NOM\_COMPLET).
  + ID\_Gabarit\_Email\_Confirmation\_[LANGUE] : (Optionnel) ID d'un Google Doc pour surcharger l'e-mail de confirmation par défaut pour une langue donnée.
  + Repondant\_Quand : Définit le délai d'envoi des résultats au répondant. Les valeurs (Immediat, 4h, 24h, etc.) proviennent de sys\_Options\_Parametres.
  + Email\_Alias : (Optionnel) Permet de spécifier une adresse d'expédition alternative configurée dans votre Gmail.
  + Colonnes ID\_Sheet\_Cible, ID\_Formulaire\_Cible, Lien\_Formulaire\_Public : Remplies automatiquement par le [MOTEUR].
* **Onglet sys\_ID\_Fichiers** : Table de correspondance centrale pour les ID des fichiers critiques.
  + ID\_BDD : L'ID de la base de données.
  + ID\_TEMPLATE\_TRAITEMENT\_V2 : L'ID du modèle de Kit de Traitement.
  + ID\_GABARIT\_CONFIRMATION\_[LANGUE] : L'ID du Google Doc servant de gabarit de confirmation **par défaut** pour chaque langue.
* **Onglet sys\_Options\_Parametres** : Alimente les menus déroulants de l'interface de configuration.

#### Fichier [BDD]V2 Tests & Profils

C'est le cerveau du système.

* **Onglet Questions\_[Type]\_[Langue]** : Stocke les questions.
  + ID : Identifiant unique et invariant de la question.
  + Paramètres (JSON) : **Le champ le plus important**. Contient la logique de calcul de la question (mode, options, profils, valeurs).
    - Exemple de JSON pour un QCU : {"mode":"QCU\_CAT", "options":[{"libelle":"Option A", "profil":"ROUGE", "valeur":1}, {"libelle":"Option B", "profil":"BLEU", "valeur":1}]}
* **Onglet Profils\_[Type]\_[Langue]** : Stocke les textes descriptifs pour chaque profil de résultat.
* **Onglet sys\_Composition\_Emails** : Le pilier du moteur d'assemblage d'e-mails. Chaque ligne est une "brique" de contenu (Sujet, Introduction, Document, Ligne\_Score...). Le script filtre ces briques en fonction du type de test, de la langue, du niveau de l'e-mail et du profil du répondant pour construire l'e-mail final.
* **Onglet traductions** : Centralise les chaînes de texte statiques (PREFIXE\_COPIE\_EMAIL, etc.). Une clé manquante pour une langue peut empêcher l'envoi d'un e-mail.
* **Onglets Obsolètes** : Pour le moteur universel, l'onglet sys\_PiecesJointes n'est plus utilisé. La logique des pièces jointes est désormais gérée directement dans sys\_Composition\_Emails avec l'élément de type Document.

### 3.2. Fonctionnalités Clés Pilotées par les Données

* **Les Blocs Méta Dynamiques** : L'interface de configuration (FormulaireUI.html) lit dynamiquement l'onglet Questions\_META\_FR de la BDD pour afficher les questions standards sous forme de cases à cocher. Cela permet d'ajouter de nouvelles questions standards sans modifier le code.
* **L'Envoi d'E-mail Différé** : Cette fonctionnalité est entièrement pilotée par la colonne Repondant\_Quand dans Paramètres Généraux. Si la valeur n'est pas "Immediat", le script TraitementReponses.gs appelle programmerEnvoiResultats() du fichier GestionTriggers.gs, qui crée un déclencheur temporel pour exécuter l'envoi plus tard.

## Partie 4 : Architecture des Scripts, Maintenance et Dépannage

### 4.1. Cartographie des Scripts

* **Scripts du [MOTEUR]** (01\_Moteur\_Script\_250822.txt)
  + MoteurV2.js : Contient la fonction principale lancerDeploiementComplet().
  + UtilsV2.js : Fonctions utilitaires pour lire la configuration et créer les items du formulaire.
  + InterfaceV2.js : Gère la création du menu utilisateur.
* **Scripts du [TEMPLATE]** (04\_Templates\_Script\_250822.txt)
  + TraitementReponses.gs : Le cœur logique du kit. Contient onFormSubmit(), traiterLigne() et assemblerEtEnvoyerEmailUniversel().
  + Utilities.gs : Boîte à outils pour lire la configuration centrale, les traductions, etc.
  + Logique\_Universel.gs : Le moteur de calcul qui interprète le Paramètres (JSON) pour déterminer les scores.
  + GestionTriggers.gs : Gère la création et l'exécution des envois d'e-mails différés.
  + Menu.js, RetraitementUI.html, DialogueLigne.html : Gèrent l'interface de retraitement.

### 4.2. Maintenance et Dépannage

* Procédure de Mise à Jour d'un Kit  
  Lorsqu'une modification est apportée aux scripts du [TEMPLATE], elle n'est pas automatiquement propagée aux Kits déjà générés. Pour mettre à jour un kit existant, il faut :
  1. Ouvrir le projet Apps Script du [TEMPLATE].
  2. Copier le contenu du ou des fichiers mis à jour.
  3. Ouvrir le projet Apps Script du Kit à mettre à jour.
  4. Coller le nouveau code en remplacement de l'ancien.
* Le Framework de Débogage  
  Pour analyser un problème sur un kit spécifique :
  1. Ouvrez l'éditeur de script du Kit de Traitement à analyser.
  2. Dans le fichier TraitementReponses.gs, passez la constante MODE\_DEBUG\_ACTIF à true.
  3. Relancez le traitement (via le formulaire ou l'interface de retraitement).
  4. Consultez les journaux détaillés dans le menu **Exécutions** de l'éditeur de script.
  5. **Important** : N'oubliez pas de repasser la constante à false une fois le débogage terminé.
* **FAQ des Erreurs Courantes**
  + **Problème : Une pièce jointe n'est pas envoyée.**
    - **Cause la plus probable** : Un caractère invisible (saut de ligne) dans la cellule de l'ID du document dans sys\_Composition\_Emails.
    - **Solution** : Le code a été renforcé avec .trim(), mais il est bon de vérifier la propreté des données.
  + **Problème : L'e-mail de copie pour le formateur/patron n'est pas reçu.**
    - **Cause la plus probable** : La clé PREFIXE\_COPIE\_EMAIL est manquante dans l'onglet traductions de la BDD.
    - **Solution** : Ajouter la clé et sa traduction pour toutes les langues.
  + **Problème : Les scores sont à zéro ou le profil est incorrect.**
    - **Cause la plus probable** : Incohérence entre le libellé de la réponse dans le formulaire et le libelle correspondant dans le Paramètres (JSON) de la question dans la BDD.
    - **Solution** : S'assurer que les textes correspondent parfaitement, notamment pour les tests multilingues.
  + **Problème : L'interface de retraitement ne se lance pas.**
    - **Cause la plus probable** : Les scripts du Kit sont obsolètes.
    - **Solution** : Mettre à jour les fichiers Menu.js, RetraitementUI.html et DialogueLigne.html du Kit avec les dernières versions du [TEMPLATE].