

LE 4GL DU 21^{ÈME} SIÈCLE

Des écrans mode texte aux smartphones



Au sommet de sa gloire, le langage de programmation IBM® Informix® 4GL a dominé les secteurs de la finance, des géants du commerce, de la production industrielle et de l'administration en raison de sa capacité à créer des applications professionnelles fiables de très haut niveau et ceci rapidement et à moindre coût.

Au milieu des années quatre-vingt-dix, l'avènement du client-serveur avec les écrans graphiques, et les langages de programmation orientés objet ont mis fin à cette gloire, même si ces tendances n'ont jamais permis d'atteindre la productivité des développeurs 4GL à ce jour.

Néanmoins, dans un monde où les budgets informatiques sont constamment sous pression, le 4GL reste pertinent et est toujours le moyen le plus productif pour développer des applications de gestion.





Au cours des 30 dernières années, les avancées technologiques ont révolutionné notre façon de travailler. Cette période n'a pas permis à IBM® de moderniser l'Informix® 4GL. Four Js a comblé cette lacune en créant Genero® qui est une réécriture totale du compilateur et de la machine virtuelle en utilisant les dernières méthodes techniques tout en restant compatible à 100% avec l'existant.

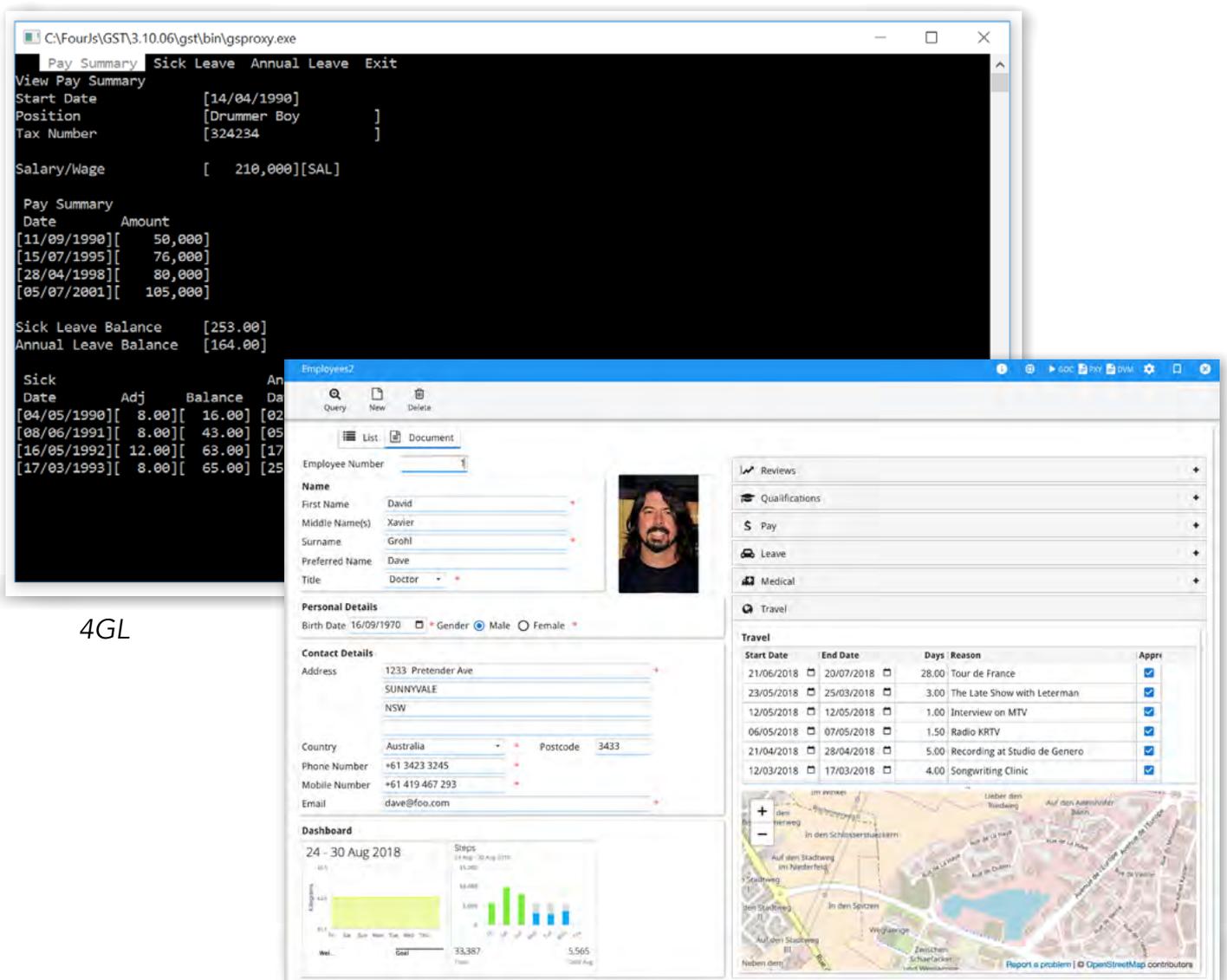
Genero permet de conserver d'une part les énormes investissements faits pour la conception des programmes ces dernières décennies et d'autre part de préserver le savoir-faire métier des équipes de développement.

Genero est l'environnement de développement le plus portable, le plus évolutif du marché lequel permet aux applications d'évoluer au rythme des tendances technologiques, avec un minimum d'efforts et de complexité.

Avec Genero, les applications 4GL s'exécutent partout - indépendamment du client, de la base de données ou du système d'exploitation - et ceci avec peu ou pas de modifications. Que ce soit sur un écran texte, sur un ordinateur portable MacOS® ou MS® Windows, dans un navigateur ou sur un smartphone, les applications s'adaptent au moment de l'exécution à l'environnement et ne doivent être développées qu'une seule fois avec un seul et même code programme.

Cela explique pourquoi Genero continue de promouvoir la cause du 4GL et de prospérer en tant que pilier de tant de systèmes d'information critiques que gèrent dans le monde entier des grandes entreprises et des institutions gouvernementales.

Genero protège des décennies d'investissement en code et en compétences.



4GL

Genero

Depuis les années 1980, avec les écrans en mode texte, le 4GL n'a jamais évolué et n'a jamais offert un environnement contemporain à ses utilisateurs avec pour conséquence de ne pas prendre en charge les fonctionnalités ergonomiques fournies par les clients modernes d'aujourd'hui : MS Windows, MacOS, navigateurs ou appareils mobiles.

L'absence d'onglets, de widgets et de barres d'outils dans le 4GL, ont conduit les programmeurs à intégrer autant d'informations que possible sur un seul écran divisé en fenêtres « maître/détail ».

Genero permet de moderniser ce fouillis, rend l'utilisation et le rendu des applications indiscernables avec les plus ergonomiques du marché d'aujourd'hui.

Genero est unique dans sa capacité à combiner le code écrit il y a 25 ans avec du code qui exploite les tous derniers paradigmes informatiques.



UNE ERGONOMIE CONTEMPORAINE

Les fenêtres contextuelles du 4GL fournissent un mécanisme permettant d'accéder à plus d'informations, mais les utilisateurs doivent quitter la fenêtre avant de pouvoir en utiliser une autre. Une autre restriction concerne les champs de saisie où les utilisateurs peuvent aller d'un champ à l'autre, mais uniquement de manière séquentielle et sont limités à un dialogue à la fois. Les applications 4GL sont inutilisables avec les navigateurs et le bouton « Retour » fait perdre le contexte. En conséquence, ils sont lourds et frustrants à utiliser pour les utilisateurs ayant l'habitude d'utiliser des navigateurs. Genero résout tous ces problèmes avec une technologie de rendu à la pointe de l'état de l'art.



INTEROPÉRABILITÉ

4GL est également notoirement introverti. Il se parle très bien à lui-même et ignore le monde extérieur. Ce n'est pas parce qu'il ne veut pas communiquer - tout simplement il ne sait pas comment faire. Les applications modernes, quelle que soit le langage de programmation avec lequel elles ont été écrites, inter-opèrent via les services Web SOAP et RESTful à l'aide de structures de données XML et JSON.

Ces fonctionnalités ont été intégrées à Genero pour rendre les applications grégaires - capables de communiquer de manière transparente avec le monde extérieur et surtout d'être utilisées facilement sans connaissance requise des protocoles sous-jacents.



SÉCURITÉ

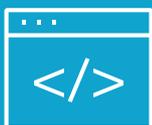
Les applications exposées au monde extérieur sont extrêmement vulnérables aux attaques. Il est donc essentiel de développer des applications sécurisées qui s'intègrent parfaitement dans ce monde complexe.

Genero fournit des classes et des méthodes qui effectuent la cryptographie, les algorithmes de calcul de signature, l'authentification, la certification et l'authentification unique (SSO) afin que les applications puissent interagir avec les médias sociaux et d'autres systèmes orientés Internet.



AMÉLIORATIONS DU LANGAGE DE PROGRAMMATION

Le langage a été amélioré pour incorporer des fonctionnalités telles que les paramètres nommés, la transmission des paramètres de fonction par référence, la gestion des événements, l'internationalisation et une interface Java®. Une classe de méthodes, rendant abstrait les systèmes d'exploitation pour la gestion des systèmes de fichiers, améliore considérablement la portabilité des applications.



UN ENVIRONNEMENT DE DÉVELOPPEMENT INTÉGRÉ ET DES OUTILS DE TESTS

Tout langage de développement moderne nécessite un environnement riche pour créer efficacement des applications. Genero Studio, Genero Report Writer et Genero Ghost Client ont pour objectif d'améliorer la productivité des développeurs et de tester les applications pour en vérifier la validité, la fiabilité, les performances et l'évolutivité.



ARCHITECTURE

Genero libère le développeur des restrictions du 4GL – notamment celle d'être uniquement monobase, de fonctionner avec le seul système d'exploitation Unix® et de n'avoir qu'un format d'affichage alphanumérique de 25 lignes de 80 colonnes. En libérant le développeur de ces contraintes, une nouvelle vie peut être insufflée aux applications qui reviennent au goût du jour et le resteront dans le futur grâce à un seul outil de développement qui utilise un seul code source. L'architecture rend les applications portables, sécurisées et évolutives à des milliers d'utilisateurs simultanés sans la nécessité d'implémenter un middleware supplémentaire et coûteux.



UNE ERGONOMIE CONTEMPORAINE

Il est possible de recompiler 'en l'état' une application 4GL avec Genero en utilisant une interface utilisateur textuelle (TUI) et de continuer à développer comme d'habitude en mode 'ASCII'. Alternativement, la recompilation en «mode traditionnel» procure auto-magiquement une interface graphique, des touches de fonction logicielles, des barres de défilement et permet l'utilisation d'une souris sans modifier le code existant.

Cette approche est rapide, mais l'interface utilisateur conserve les contraintes et restrictions du mode texte ASCII. Il serait dommage de ne pas utiliser les avantages et les extensions GUI (Graphical User Interface) de Genero qui donneraient une ergonomie contemporaine aux applications.

Le changement le plus important apporté au langage dans Genero est sans doute l'introduction du mot-clé LAYOUT pour les fichiers «.per». Lorsqu'il est utilisé à la place de SCREEN, l'écran sera interprété comme un écran graphique «normal» activé par un sur-ensemble de la syntaxe de description d'écran.

Dans de nombreux cas, le plus grand impact visuel peut être obtenu en modifiant simplement les fichiers «.per» les plus importants. Genero étend la syntaxe d'écran dans le fichier «.per» avec les éléments graphiques abstraits ci-dessous qui s'affichent en mode natif sur Windows, Linux®, MacOS, les navigateurs et les API mobiles.

Les widgets graphiques définissent la manière dont les valeurs sont affichées et modifiées sur l'écran :

- **BUTTON** – zone cliquable sur l'écran qui déclenche une action,
- **BUTTONEDIT** – zone éditabile et cliquable pour déclencher une action,
- **CHECKBOX** – affiche une ou plusieurs cases à cocher,
- **COMBOBOX** – affiche une liste de valeurs défilante,
- **DATEEDIT** – affiche un calendrier permettant de sélectionner une date,
- **DATETIMEEDIT** – affiche la date et l'heure permettant une sélection,
- **EDIT** – affiche une zone éditabile ou une valeur par défaut,
- **IMAGE** - affiche une image,
- **LABEL** - affiche une étiquette,
- **PROGRESSBAR** - affiche une barre de progression,
- **RADIOGROUP** – sélection d'une valeur parmi de multiples choix,
- **SLIDER** – affiche un curseur permettant d'augmenter ou de diminuer une valeur,
- **SPINEDIT** – Permet d'ajuster une valeur sur un tambour,
- **TEXTEDIT** – affiche une zone de texte éditabile,
- **TIMEEDIT** – affiche et permet de sélectionner une heure.

Des conteneurs pour définir la disposition des widgets graphiques les uns par rapport aux autres

- **GRID** - positionne les widgets avec des paramètres X, Y,
- **SCROLLGRID** - organise plusieurs instances d'un GRID,
- **TABLE** - organise sur une ou plusieurs lignes une structure de type table,
- **TREE** - similaire à TABLE mais avec une structure arborescente pliable,
- **STACK** – permet d'empiler les conteneurs,
- **GROUP** - regroupe plusieurs conteneurs dans un box avec ou sans label,
- **FOLDER + PAGE** – permet d'organiser et d'afficher les conteneurs fils sous forme d'onglets, où en forme d'accordéon repliable,
- **VBOX** – empile les conteneurs verticalement,
- **HBOX** - organise les conteneurs horizontalement.

Les composants Web – permettent l'intégration des technologies HTML tierces existantes (HTML / JS / CSS) et de créer des composants d'interface utilisateur personnalisés. Exemples d'utilisation : intégration de vidéos, documents HTML, cartes, graphiques, édition de texte enrichi, calendrier, capture de signature, tableaux riches, cartes d'image / écrans tactiles POS, authentification SSO. Les composants Web prédéfinis permettent l'édition de texte enrichi, la sélection d'images multiples et le rendu SVG.

Action defaults centralise la définition du texte, des commentaires, des images, des accélérateurs et les propriétés des actions,

Top menus (Menu bar) – menus principaux (barre de menus) - un agencement structuré des options de menu,

Toolbars – barre d'outils librement positionnable,

Parallel dialogs – permet d'avoir dans un même écran 2 dialogues actifs exécutant différents codes,

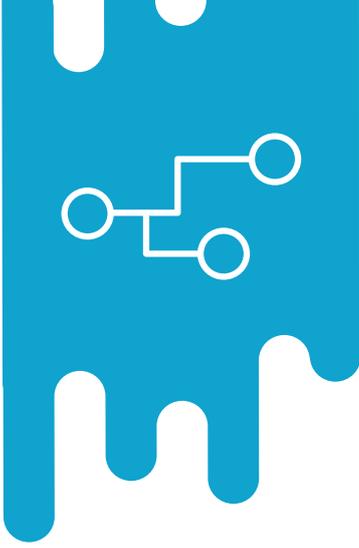
Multiple dialogs – regroupe des instructions d'interaction (MENU, INPUT, INPUT ARRAY, DISPLAY ARRAY, CONSTRUCT) en une boîte de dialogue active facilitant la navigation de l'utilisateur,

Dynamic dialogs – construit des dialogues (INPUT, CONSTRUCT, DISPLAY ARRAY, INPUT ARRAY, DIALOG, ...) qui sont définis dynamiquement au moment de l'exécution plutôt qu'à la compilation. Cela permet l'utilisation d'un code générique avec une structure de données variable. Utilisé conjointement avec la création dynamique des écrans au moment de l'exécution et l'utilisation de curseurs dynamiques de base de données à partir de la classe base.Sqlhandle,

Multiple selects – possibilité de sélectionner plusieurs lignes dans une liste pour effectuer des actions telle que la suppression ou l'édition,

Focus on field – permet de sélectionner une cellule dans un tableau,

Drag and drop – permet de glisser-déposer certains objets les uns sur les autres pour effectuer une action spécifique pour une application ou entre plusieurs.



INTEROPÉRABILITÉ

Genero a évolué au cours des 25 dernières années pour inclure les dernières API et structures de données pour interagir avec d'autres systèmes. Cela inclut les structures de données les plus populaires : XML et JSON ainsi que les méthodes les plus courantes : les services Web, SOAP et RESTful.

Structure des données

XML

Genero fournit un ensemble complet de classes et de méthodes pour travailler avec des documents XML. Cela inclut la prise en charge de DOM (Document Object Modeling) et StAX (Streaming API pour XML), ainsi que des modèles de sérialisations et de transformations XSLT.

JSON

Genero fournit des classes pour travailler avec JSON (JavaScript Object Notation)

Méthodes

Services Web – Genero fournit des bibliothèques pour créer des services Web, coté serveur, comme fournisseur de services Web ou comme client du service Web.

SOAP – le fichier WSDL, qui fournit les détails de l'interface, constitue l'élément essentiel. A partir d'une fonction 4GL existante comportant des paramètres d'entrée et de sortie, Genero crée automatiquement ou publie un service Web basée sur cette fonction.

RESTful – plus léger que SOAP permet de passer des messages (JSON, XML en général) dans un format au choix.

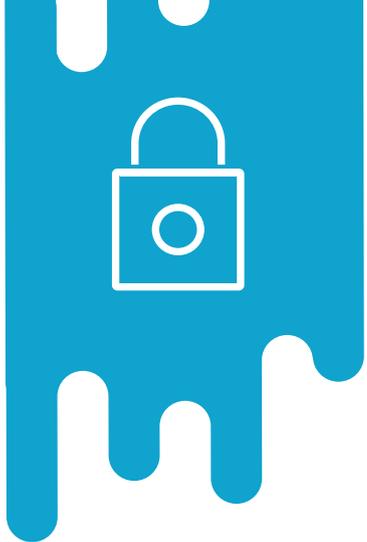
File – l'utilisation de ces classes permet d'éliminer les bibliothèques «C».

Base.Channel – c'est un ensemble de méthodes utilisé pour lire et écrire dans des fichiers et pouvant être utilisé pour remplacer les scripts C et shell.

OS – cette classe permet de répertorier les fichiers dans des répertoires et de tester l'existence et les autorisations de ceux-ci. Cela supprime les dépendances du système d'exploitation et rend le code portable.

Sockets – des méthodes ajoutées à la classe base.Channel pour faciliter la communication entre les sockets, à la fois comme client ou serveur vers / depuis un serveur et un port donnés.

Pipes – la méthode openPipe a été ajoutée à la classe base.Channel pour faciliter la communication avec un sous-processus via un canal.



SECURITÉ

Security Class – ensemble de classes et de méthodes pour effectuer la cryptographie de base. Inclut les classes PBKDF2 et BCrypt pour le stockage et la vérification des mots de passe chiffrés. Des classes permettent d'implémenter des algorithmes de calcul tels que SHA512, MD5 (etc.), ainsi que des classes pour les algorithmes HexBinary et Base64.

Communications sécurisées – pour échanger des messages sur le Web, Genero fournit des classes pour authentifier l'origine du message, pour s'assurer qu'aucune application malveillante n'a modifié le message original et pour garantir qu'aucune application tierce n'a intercepté le message. Cela concerne les certificats, les autorités de certification, les clés privées, etc.



LES EXTENSIONS DU LANGAGE

La syntaxe du langage 4GL a été grandement améliorée pour supprimer les limites et se conformer aux pratiques de codage modernes. Voici une liste non exhaustive des améliorations apportées par Genero :

```
LET x = s.trim().getLength()
DEFINE month DICTIONARY
OF INTEGER
LET month["January"] = 1
DISPLAY month["January"]
```

Syntaxe

Mots clés nouveaux – STRING, CONSTANT, BOOLEAN, TYPE, BIGINT, TINYINT, PUBLIC, PRIVATE, NVL, IFF, SFMT, DICTIONARY, DYNAMIC ARRAY, TRY/CATCH

ON ACTION – L'instruction ON ACTION exécute un code lorsqu'un évènement est déclenché par l'utilisateur :

```
ON ACTION zoom
  plutôt que :
ON KEY(F5)
```

L'action 'zoom' est spécifiée dans le fichier d'écran .per ou dans le fichier d'action par défaut, et exécutée indépendamment de la manière dont elle a été déclenchée.

Passage des paramètres de fonction par référence – cette fonctionnalité est très utile pour traiter des structures complexes, des enregistrements et des tableaux. Les structures complexes sont transmises à la fonction sans qu'il soit nécessaire de traiter le résultat renvoyé. Une seule instance réside en mémoire permettant d'éviter deux copies (une pour fonction appelante et une pour la fonction appelée) et consommer moins de mémoire.

```
DEFINE arr DYNAMIC ARRAY OF complexDataType
```

```
# arr contient des données non transformées
CALL transform_array(arr)
# arr contient les données transformées
```

Initialisation de constantes – les constantes peuvent être initialisées simplement :

```
TYPE colorType RECORD
  red, green, blue INTEGER
END RECORD
```

```
DEFINE purple colorType = (255,0,255)
```

Case sensitivity (optionnel) - si sélectionné la syntaxe ci-dessous devient invalide pour le compilateur :

```
DEFINE foo, Foo, FOO STRING
et :
DEFINE accountCode STRING
DEFINE accountCodeLength INTEGER
  LET accountCodeLength = accountCode.getLength()
```

Paramètres nommés – suppriment la nécessité de spécifier les arguments qu’il n’est pas nécessaire de mettre dans le bon ordre lors de l’appel de fonction. Les paramètres non spécifiés sont prédéfinis avec des valeurs par défaut. Améliore la lisibilité du code.

```
CALL fgl_report_configurePDFDevice(fromPage = 1, toPage = 1)
```

Autres possibilités – exemples de syntaxe qui facilitent la migration vers Genero :

Les déclarations de type sont autorisées dans les paramètres d’appel et dans les retours de types de fonction - une manière rapide d’améliorer la sécurité du type,

```
FUNCTION add(x INTEGER, y INTEGER) RETURNS INTEGER
```

Méthode rapide de chaînage d’objets,

```
LET x = s.trim().getLength()
```

Un nouveau DICTIONNAIRE (DICTIONARY) de type de données (cf. une table de hash) utilise comme clé une chaîne de caractères pour indexer les tables. Il peut stocker tous les types de données Genero :

```
DEFINE month DICTIONARY OF INTEGER
LET month["January"] = 1
DISPLAY month["January"]
```

ANYRECORD – type d’enregistrement générique

Au moment de la compilation, seul le type d’une fonction (paramètres, valeurs de retour) est connu lors de l’utilisation d’une fonction par référence. Le type d’enregistrement générique ANYRECORD est utilisé lorsqu’une fonction possède, en entrée, un enregistrement dont la structure est inconnue.

IMPORT – permet d’importer des bibliothèques externes :

- IMPORT FGL - importe une bibliothèque 4GL
- IMPORT - importe une bibliothèque ‘C’
- IMPORT JAVA - importe une bibliothèque Java

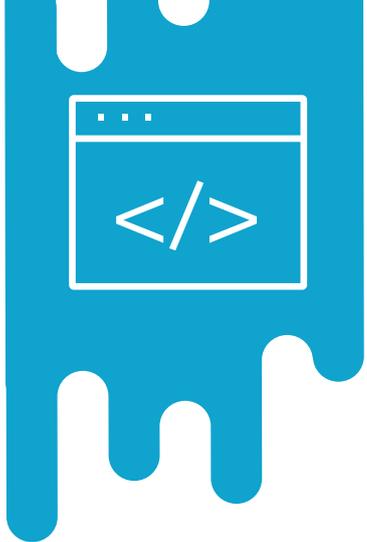
Par exemple IMPORT JAVA est utilisé pour :

- La création de fichiers MS® Word et Excel avec la bibliothèque Apache POI
- Télécharger des vidéos vers YouTube®

Internationalisation - les applications écrites en Genero peuvent être utilisées dans le monde entier grâce au support UTF-8. Le code est indépendant du jeu de caractères.

Une **chaîne localisée** est un texte de l’application qui est stocké dans un dictionnaire extérieur au code source standard. Cela facilite la maintenance du texte traduit et permet l’utilisation d’une traduction spécifique au moment de l’exécution sans avoir à recompiler.

Un **environnement d’application** détermine le choix du dictionnaire régional qui est utilisé pour un utilisateur donné. Ainsi, une même instance d’application peut prendre en charge plusieurs langues simultanément. L’écriture de droite vers la gauche est également supportée pour les régions arabes et les sémantiques de longueur BYTE et CHARACTER sont prises en charge pour l’Asie.



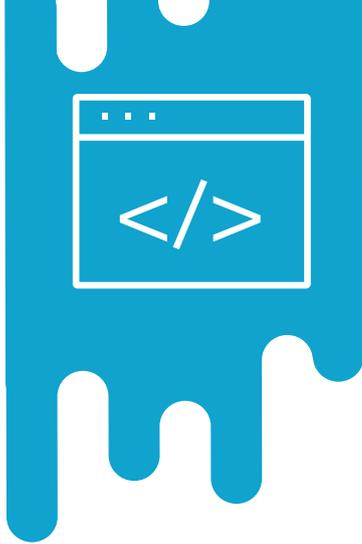
UN ENVIRONNEMENT DE DÉVELOPPEMENT INTÉGRÉ (GENERO STUDIO)

Genero Studio est un environnement de développement intégré (IDE) conçu pour améliorer la productivité des développeurs. Il permet de déboguer simultanément une application développée pour PC, MAC, LINUX, navigateurs et les appareils mobiles.

Il comprend un éditeur de code sensible au langage qui prend en charge la complétion du code, la vérification syntaxique en temps réel, la mise en évidence des mots clés 4GL ainsi que d'autres formats de fichiers.

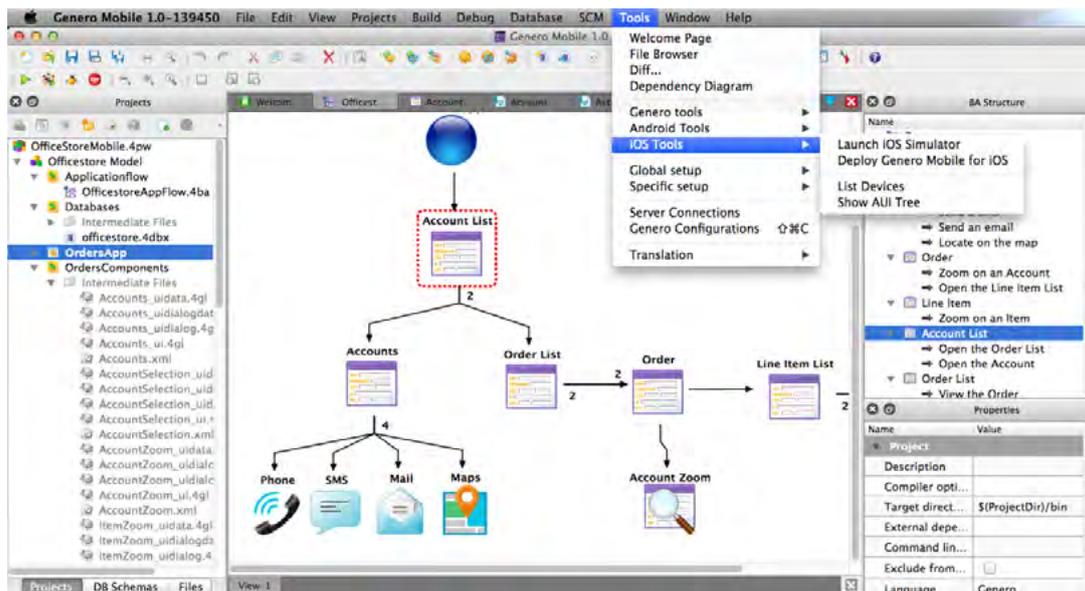
Il comprend également un débogueur graphique, un concepteur d'écrans graphiques, un designer d'édition graphique, un manager de projet pour « créer » en équipe des applications complexes, un générateur de schéma de base de données et d'application, un méta-dictionnaire, un outil de modélisation visuelle, des outils de profilage et d'analyse.





LE DÉVELOPPEMENT DU CODE APPLICATIF RÉDUIT AU MINIMUM

Genero Studio Business Application Modeller est une puissante plateforme de développement rapide intégrant un générateur d'application entièrement personnalisable, basé sur des diagrammes décrivant la logique métier. Il permet l'écriture de projet d'envergure avec un cycle de développement agile permettant de simplifier drastiquement la maintenance des applications. Il est destiné aux équipes de développement soucieuses de la cohérence de l'ergonomie, de l'architecture, et des règles de codage et il permet aux développeurs de se concentrer sur la modélisation du fonctionnel de l'applicatif.



BAM - intégration native d'une application smartphone dans le module Genero Mobile

Le code généré des applications et des services web est défini dans des modèles génériques de code (template) personnalisables, ce qui garantit la cohérence et la certitude d'avoir un code structuré et organisé en couches applicatives pour :

- Le stockage des données,
- Les services de données,
- La publication de services Web (RESTful, SOAP),
- La définition des écrans et des éditions.

Les «patterns» de codage les plus fréquents dans les applications métiers tels que les formulaires CRUD, les listes, les détails, le filtrage de données, la navigation dans les données, la gestion des accès utilisateurs concurrents, les éditions... sont inclus dans les modèles de code générique (template) par défaut. Pour les applications mobiles, l'interfaçage avec les applications natives du téléphone ou de la tablette tel que les contacts, le calendrier, la gestion des appels téléphoniques, etc. est fourni également, permettant ainsi la description de l'interface avec ces applications directement dans les diagrammes.

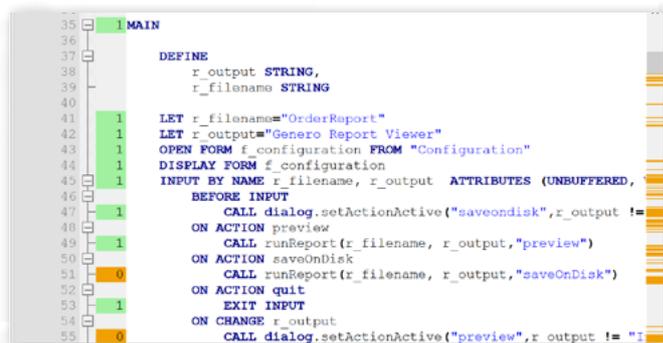
La grande majorité du fonctionnel de l'application est décrite dans les diagrammes sans nécessiter aucun codage. Le développeur fait évoluer les diagrammes pour enrichir ou modifier le comportement applicatif. Du code métier très spécifique peut néanmoins être ajouté dans des événements lorsque nécessaire.

DES PUISSANTES AIDES AU DÉVELOPPEMENT

Ces outils intégrés en standard peuvent être activés lors de la compilation ou de l'exécution depuis Genero Studio ou depuis une ligne de commande.

Profiler – profiler analyse le code et permet d'identifier les goulots d'étranglement des programmes. Il met en évidence le pourcentage de temps passé dans les fonctions et résume combien de fois les fonctions sont appelées à partir d'autres fonctions.

Code coverage – l'outil « code coverage » analyse le code source et détermine les lignes de codes inutilisées dans une application. En exécutant un programme de multiples fois avec « code coverage » activé, un compteur fournit, pour chaque ligne de code, le nombre de fois où celle-ci a été exécutée pendant le déroulement de l'application.



```
35 1 MAIN
36
37 DEFINE
38   r_output STRING,
39   r_filename STRING
40
41 1 LET r_filename="OrderReport"
42 1 LET r_output="Genero Report Viewer"
43 1 OPEN FORM f_configuration FROM "Configuration"
44 1 DISPLAY FORM f_configuration
45 1 INPUT BY NAME r_filename, r_output ATTRIBUTES (UNBUFFERED,
46
47 1 BEFORE INPUT
48   CALL dialog.setActionActive("saveondisk",r_output !=
49 1 ON ACTION preview
50   CALL runReport(r_filename, r_output,"preview")
51 0 ON ACTION saveOnDisk
52   CALL runReport(r_filename, r_output,"saveOnDisk")
53 1 ON ACTION quit
54   EXIT INPUT
55 0 ON CHANGE r_output
56   CALL dialog.setActionActive("preview",r_output != "I
```

Trace – la trace est un outil de débogage qui crée un journal, complet ou partiel selon une sélection spécifique, des appels de fonction effectués par un programme avec les arguments transmis et renvoyés.

Shared libraries – l'architecture de déploiement est une machine virtuelle qui repose sur des bibliothèques partagées, ce qui élimine le besoin de créer des machines virtuelles spécifiques.

Outils d'indentation et de modélisation

TESTS AUTOMATISÉS (GENERO GHOST CLIENT)

Le « Genero Ghost Client » permet :

- D'effectuer des séries de tests unitaires à l'aide de scénarios qui valident la logique métier,
- D'effectuer des tests de charge pour déterminer le nombre d'utilisateurs pouvant être supporté par l'infrastructure,
- De déterminer les temps de réponse de référence pour des charges d'utilisation importantes.





UN GÉNÉRATEUR D'ÉDITION (GENERO REPORT WRITER)

L'instruction REPORT est utilisée en 4GL pour produire des éditions de gestion lesquelles sont très primitives et ne correspondent plus aux standards et aux exigences d'aujourd'hui.

Genero Report Writer est un outil sophistiqué et puissant de production d'éditions diverses comme des rapports de gestion, des factures, des étiquettes d'expédition, etc. pour ne citer que ces quelques exemples.

Sa particularité réside dans sa capacité à diffuser des documents volumineux vers des imprimantes, des écrans ou même vers de nombreux formats de fichiers différents : pdf, svg, xlsx, docx, html, ps, jpg et png.

La diffusion en flux continu permet une impression immédiate de la première page lorsqu'il y a des centaines voire des milliers de pages à imprimer. Il économise des ressources système précieuses pour les documents volumineux en éliminant le besoin de créer des fichiers temporaires volumineux.

Genero Report Writer prend en charge les objets graphiques les plus usités tels que les images, les tableaux, les codes-barres, les codes QR, les graphiques, ainsi que la gestion de différents types de caractères, tailles de point et couleurs. Les mises en page sont entièrement dynamiques pour faciliter la transformation de documents d'un facteur de forme à un autre, sans qu'il soit nécessaire de retravailler la conception

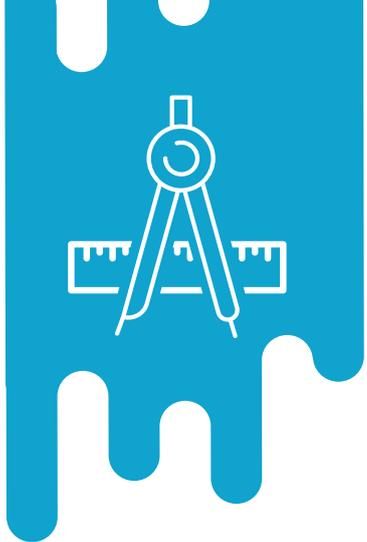
Les éditions sont créées facilement avec le « Graphical Report Designer » - un concepteur proche de WYSIWYG qui permet de modifier les formats de page sans avoir à revoir la présentation de l'édition.

La disposition de l'édition est dynamique et elle se redimensionne automatiquement quelle que soit la taille des polices, du titre, etc. et quel que soit le format de sortie sans changer le code.

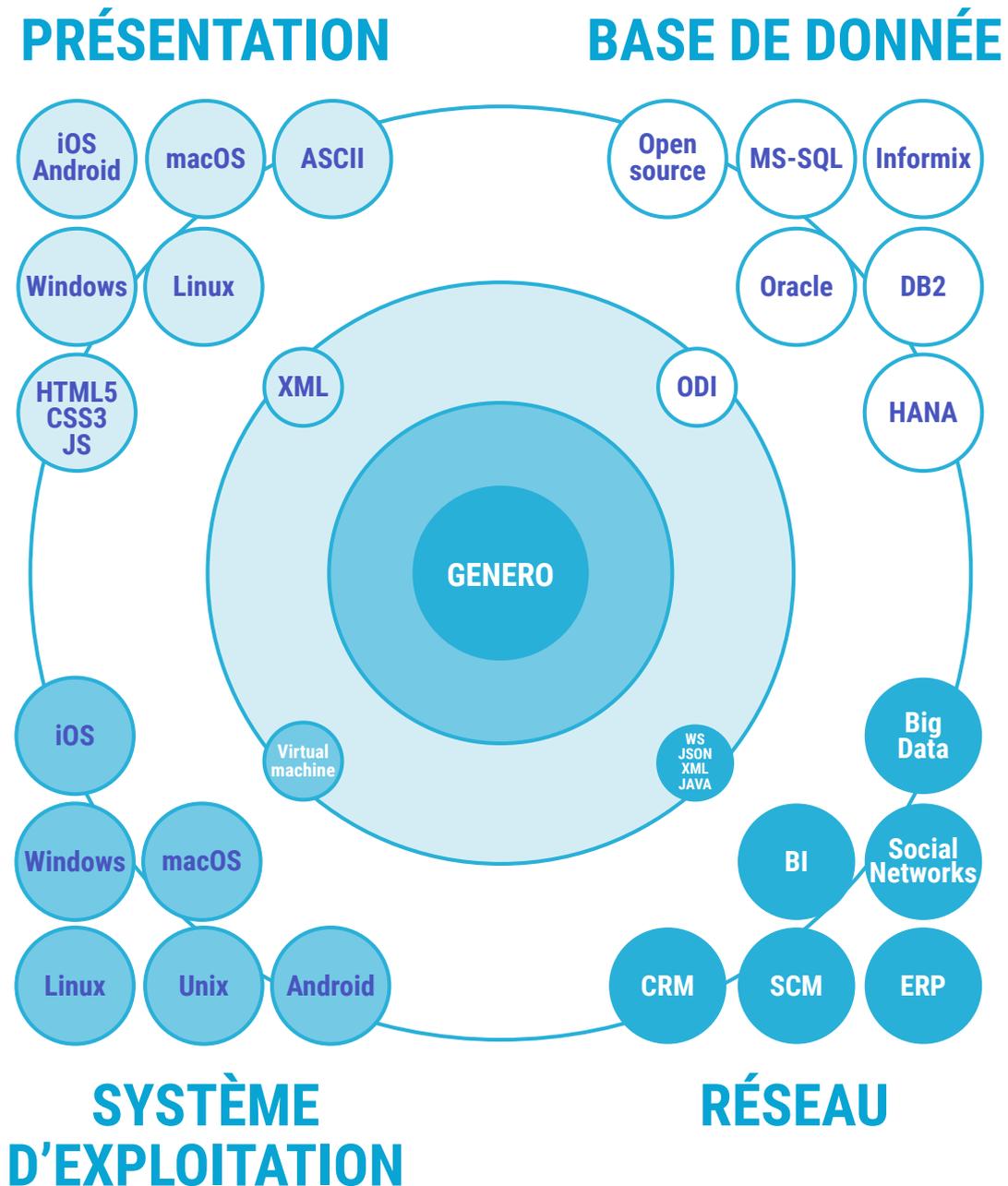
Genero Report Writer supporte des langages de programmation autres que Genero telles que Java, C, Cobol et PHP et permet ainsi l'intégration dans un réseau hétérogène.

Le mode de compatibilité 4GL

Le mode de compatibilité permet de générer automatiquement des éditions 4GL existantes en utilisant l'une des méthodes prises en charge par Genero Report Writer.



ARCHITECTURE DE GENERO



L'architecture de Genero est sécurisée, portable, évolutive pour des milliers d'utilisateurs et elle intègre son middleware évitant ainsi des dépenses supplémentaires inutiles.

NOS BUREAUX À TRAVERS LE MONDE

ÉTATS-UNIS & CANADA

Four Js Development Tools Inc.

251 O'Connor Ridge Blvd. Suite 125
Irving, Texas, 75038
États-Unis d'Amérique

Ligne gratuite : 866 314 7300 (Free)
Téléphone : (972) 893-7300
Fax: (972) 893-7304

ASIE

Four Js Development Tools Asia

Suite 210 29 Kiora Rd,
Miranda NSW, 2228 Sydney
Australie

Téléphone : +612-8004-5890

AMÉRIQUE LATINE

Four Js Development Tools
Latin America SACV.

Insurgentes Sur 1602
Col. Credito Constructor
Mexico City, DF 03940
Mexique

Téléphone : +52 (55) 1000 9160

FRANCE

Four Js Development Tools SARL.

28 Quai Gallieni
92 150 Suresnes
France

Téléphone : +33 1 41 38 86 30
Fax : +33 1 41 38 84 46

EUROPE

Four Js Development Tools Europe Ltd.

Suite 5, Rineanna House
Shannon Free Zone
Shannon V14 CA36
Co Clare, Irlande

Téléphone : +353 61 708 314
Fax : +353 61 708 315

ROYAUME-UNI

Four Js Development Tools UK LTD.

Regus House, Victory Way Admirals Park
Dartford, DA2 6QD
Royaume-Uni

Téléphone : +44 (0) 370 111 5140
Fax : +44 (0) 370 111 5154

EUROPE CENTRALE

Four Js Development Tools
Software Vertriebs GmbH

Leonhardsweg 2
82008 Unterhaching
Allemagne

Téléphone : +49 89 60815400
Fax : +49 89 60815402

ESPAGNE

Four Js Development Tools
Iberica LTDA.

Martinez Villergas 49, 1a Planta,
28027 Madrid
Espagne

Téléphone : +34 91 047 76 61
Fax : +34 91 047 76 57

ITALIE

Four Js Development Tools Italia

Via Cipriani, 2
42124 Reggio Emilia
Italie

Téléphone : +39 (0)522 420786
Fax : +39 (0)522 420768

www.4js.com

Marques déposées

- ® Four Js, Genero et ses logos sont des marques déposées de Four Js Development Tools Europe Ltd.
- ® Mac, macOS and IOS sont des marques déposées de Apple Corps.
- ® IBM, Informix and DB2 sont des marques déposées de IBM Corporation.
- ® Microsoft, MS Windows, and MS SQL sont des marques déposées de Microsoft Corporation.
- ® Java and Oracle sont des marques déposées de Oracle Corporation.
- ® Linux est une marque déposée appartenant à Linus Torvalds.
- ® SAP HANA est une marque déposée de SAP SE en Allemagne et autres pays.
- ® UNIX est une marque déposée de The Open Group for the United States and other countries.
- ® You Tube et Android sont des marques déposées de Alphabet Corp.

© 1995-2018 tous droits réservés.

