



## COMMENT GARDER L'ISO FONCTIONNALITE

### FONCTIONNALITES

#### Nouveautés

- WebServices Rest : contrôle de la cohérence dans GTPWD
- WebServices Rest : contrôle des types d'authentifications autorisés
- WebServices Rest : Génération d'APIKEY
- WebServices Rest : sécurité CSRF
- WebServices Rest : cookies sécurisés
- WebServices Rest : préciser un établissement de connexion
- WebServices Rest : Service générique de recherche
- WebServices Rest : Service générique de recherche "auto généré"

#### Modifications

- ECFPDF : Concaténation des fichiers résultats PDF
- GTENF : Entités par utilisateur
- TTAU : Génération transactions et indicateurs par utilisateur
- ETAU : Transactions et indic. accessibles par utilisateur
- GTSER : Symboles
- TDBRQ : Traitement de duplication des briques
- MVAR : Propagation des variantes
- DUSR : Duplication d'un utilisateur

### MODULES

#### Nouveautés

- QIN - XLinks : Classes autorisées dans les directives
- QIN - XLinks : Concaténation simplifiée
- QIN - XLinks : Fichiers dans les spools des jobs
- QIN - XLinks : Fichiers basés
- QIN - XLinks : Console d'exécution
- QIN - XLinks : Verrous
- QIN - XLinks : Tris
- QIN - XLinks : Extraction des interfaces basées
- QIN - XLinks

#### Modifications

- QIN - XLinks : Editeur de requêtes
- QIN - XLinks : Changer le nom de balise
- QIN - XLinks : Sauvegarde des interfaces
- QIN - XLinks : Rechercher et remplacer
- RGPD : Possibilité de gérer l'anonymisation pour une adresse de tiers, sans traiter sur le tiers
- QIN - XLinks
- QIN - XLinks

## **Comment garder l'iso fonctionnalité**

### **WebServices Rest et API d'authentification**

**Pour des raisons de sécurité, les API d'authentification utilisant le verbe http "GET" ne sont plus disponibles.**

**"GET" /security/authentication est remplacée par l'API "POST" /security/authentication ou "POST" /security/authenticationForm**

**"GET" /security/changePassword est remplacée par l'API "POST" /security/changePassword**

**"GET" /security/lostPassword est remplacée par l'API "POST" /security/lostPassword**

**Si ces méthodes "GET" étaient utilisées, il faut les remplacer par leur équivalent "POST".**

# Fonctionnalités

## Nouveautés

### WebServices Rest : contrôle de la cohérence dans GTPWD

Afin d'assurer un fonctionnement sécurisé des WebServices Rest, il est nécessaire de définir une clef pour la gestion des mots de passe (GTPWD). Il doit exister une valeur pour le type JWT et l'utilisateur "mobile". Si cela n'est pas le cas, l'instance WebService sur le serveur Web ne peut pas démarrer correctement.

Désormais, si cette clef n'est pas définie, une clef aléatoire est créée au moment du démarrage du serveur Web. De plus, toujours lors du démarrage, si on constate qu'il existe plusieurs clefs, une erreur est affichée dans le fichier de logs du serveur Web, qui signale un problème lors de démarrages.

### WebServices Rest : contrôle des types d'authentifications autorisés

La propriété **qualiac.security.authmode** permet de préciser les types d'authentifications autorisés sur l'instance.

Une combinaison de plusieurs valeurs peut être définie :

- USER, pour autoriser une authentification classique via couple user/password
- INHOUSEAPIKEY, pour utiliser une APIKEY "maison" (hors passeport)
- PASSEPORTAPIKEY, pour utiliser une APIKEY fournie via passeport (offre Cegid Immobilisation Ultimate uniquement)
- OIDC, pour passer via une authentification openId Connect (offre Cegid Immobilisation Ultimate uniquement)
- ALL, pour autoriser l'ensemble des types précédents.

On peut combiner plusieurs valeurs, en les séparant par des |. Par exemple USER|OIDC

### WebServices Rest : Génération d'APIKEY

Le principe est de pouvoir s'authentifier aux services sans avoir à passer un couple user/password. Cela peut s'avérer indispensable dans le cas de dialogue machine-to-machine et ne pas avoir à utiliser un vrai compte (notamment si utilisation de LDAP), en utilisant plutôt un compte dit "de service".

#### **Explications**

Nous fournissons donc la possibilité d'appeler le service d'authentification avec une clef. Cette clef (ApiKey) doit en premier lieu être créée et fournie à la partie cliente.

Elle doit être précieusement conservée et protégée par cette partie.

Pour la créer, il faut être authentifié et utiliser le service misc/apikey.

On précise l'utilisateur "modèle" (qui peut être différent de l'utilisateur connecté courant) qui sera associé à l'API key, ainsi que la durée de validité de la clef (limitée à 365 j max, il faudra donc penser à la renouveler tous les ans au minimum).

**L'apikey ne peut pas être retrouvée, il faut donc la noter dès le retour du service dans un endroit sécurisé.**

**Seul un utilisateur ayant les droits (GUDR) de créer un user peut utiliser le WebService misc/apikey.**

De plus, si la propriété **qualiac.security.authmode** ne prévoit pas de gérer INHOUSEAPIKEY, cela veut dire que nos WebServices ne peuvent fonctionner avec ce type d'authentification. Il ne sera pas possible ni d'utiliser ce service ni de créer une ApiKey.

Pour s'authentifier avec cette apikey, il suffira de la passer dans le paramètre "apikey" des services d'authentification, en laissant le champ "password" vide.

## **WebServices Rest : sécurité CSRF**

Il est possible d'activer une protection contre les attaques de type CSRF.

### **Explications**

Pour cela, il faut positionner la propriété `qualiac.security.anticsrf=true`  
De ce fait, lors de l'authentification, un jeton "anti CSRF" va être positionné en retour. Il sera transmis selon le type d'authentification choisie, soit cookie, soit header.  
Si c'est un cookie, il n'est pas `httpOnly`, afin que le client puisse le lire. Par contre il est bien positionné sur un domaine pour que seul le domaine émetteur puisse le lire.  
De plus, si on est en mode "secure" du cookie (propriété `qualiac.security.securecookie=true`), le nom de ce dernier est préfixé par "`__Host-`" afin d'utiliser la fonctionnalité des suffixes de cookie. Cela permet de donner l'accès en lecture uniquement sur le domaine qui à positionné le cookie.

Ensuite, lors des appels des services fonctionnels, il faudra impérativement passer ce "aCsrfToken" en tant qu'en tête de la requête (Nom du header = `com.cegid.auth.csrf`).

## **WebServices Rest : cookies sécurisés**

Pour activer l'envoi de cookies uniquement en mode https, il faut positionner `qualiac.security.securecookie=true`

## **WebServices Rest : préciser un établissement de connexion**

L'attribut "connectionEntity" (optionnel) est ajouté lors de l'authentification. Il permet de positionner un établissement de connexion.

## **WebServices Rest : Service générique de recherche**

Les services

`POST /api/foundations/genericQuery/{queryName}`

ou

`POST /api/foundations/genericQueryAsStream/{queryName}`

permettent d'exécuter une requête Sql prédéfinie dans le produit et d'en retourner le résultat.

Les grands principes sont les suivants :

- Ecriture du select à lancer via les gestions de Workflow Information Manager (GTUREQ, etc.)
- possibilité de prédéfinir des conditions dynamiques
- format en sortie JSON ou CSV

## **WebServices Rest : Service générique de recherche "auto généré"**

Le principe est de pouvoir lancer une requête sur n'importe quelle table sans avoir à écrire un quelconque select via l'interface Cegid XRP Ultimate. C'est donc l'appel du service qui doit préciser quelle est la table à lire, quels sont les critères à appliquer, quelles sont les colonnes à retourner, etc.

### **Explications**

Il est donc possible de lancer des select sans grands niveau de complexité SUR UNE TABLE UNIQUE. Aucune jointure n'est possible.

L'aspect sécurité est particulièrement pris en compte dans cette fonctionnalité

Les services concernés sont

POST /api/foundations/autoGeneratedQuery/{tableName}

et

POST /api/foundations/autoGeneratedQueryStream/{tableName}

Les grands principes sont les suivants :

- le nom de la table souhaitées est passé en paramètre de l'URL
- des critères peuvent être passés sur chaque colonne
- on peut préciser les colonnes ramenées
- on peut limiter le nombre d'enregistrements ramenés
- format JSON ou CSV en sortie

## Modifications

### **ECFPDF : Concaténation des fichiers résultats PDF**

Il est possible d'insérer le fichier généré dans la gestion électronique de documents.

#### ***Explications***

Pour plus de détail voir la documentation de la soumission ECFPDF

#### ***Transactions concernées (ou liste des modifications)***

**ECFPDF - Concaténation des fichiers résultats PDF (Transaction GTECPDF)**

### **GTENF : Entités par utilisateur**

Prise en compte de la sécurité lors de l'accès aux WebServices.

#### ***Explications***

Lors de l'appel à un service, si la globale RWS\_SYSSEC est à "O", le système peut contrôler que l'utilisateur a le droit d'appeler ou non ce service.

#### ***Transactions concernées (ou liste des modifications)***

**GTENF - Entités par utilisateur (Transaction GTIENF)**

#### ***Mise en place***

Pour la mise en place de la sécurité sur les WebServices, il faut :

- positionner à "O" la globale RWS\_SYSSEC ;
- référencer le service dans GTSVC si celui-ci ne l'est pas ;
- autoriser ce service durant une période à un utilisateur et/ou profil via la transaction GTENF ;
- lancer le traitement TTAU pour prendre en compte ce paramétrage en générant les droits d'accès.

Pour plus d'information voir : La sécurité.

### **TTAU : Génération transactions et indicateurs par utilisateur**

Prise en compte de la sécurité lors de l'accès aux WebServices.

#### ***Explications***

Lors de l'appel à un service, si la globale RWS\_SYSSEC est à "O", le système peut contrôler que l'utilisateur a le droit d'appeler ou non ce service.

### ***Transactions concernées (ou liste des modifications)***

#### **TTAU - Génération des accès par utilisateur (*Transaction GTTAU*)**

### ***Mise en place***

Pour la mise en place de la sécurité sur les WebServices, il faut :

- positionner à "O" la globale RWS\_SYSSEC ;
- référencer le service dans GTSVC si celui-ci ne l'est pas ;
- autoriser ce service durant une période à un utilisateur et/ou profil via la transaction GTENF ;
- lancer le traitement TTAU pour prendre en compte ce paramétrage en générant les droits d'accès.

Pour plus d'information voir : La sécurité.

## **ETAU : Transactions et indic. accessibles par utilisateur**

Prise en compte de la sécurité lors de l'accès aux WebServices.

### ***Explications***

Lors de l'appel à un service, si la globale RWS\_SYSSEC est à "O", le système peut contrôler que l'utilisateur a le droit d'appeler ou non ce service.

De plus, pour corriger un problème avec le calcul des dates, le fonctionnement de l'édition a été entièrement revu.

### ***Transactions concernées (ou liste des modifications)***

#### **ETAU - Simulation de la génér. des accès par utilisateur (*Transaction GTETAU*)**

## **GTSER : Symboles**

Prise en compte des symboles définis dans GTSER lors de la saisie des valeurs par défaut dans les transactions.

### ***Explications***

Lors de la saisie des valeurs par défaut dans les transactions, il est désormais possible de créer de nouveaux symboles venant se rajouter à la liste des symboles utilisables en mode paramétrage.

### ***Transactions concernées (ou liste des modifications)***

#### **GTSER - Symboles (*Transaction GTSER*)**

## **TDBRQ : Traitement de duplication des briques**

Traitement permettant de dupliquer les briques et optionnellement les éléments liés à cette brique.

### ***Explications***

Ce traitement permet de dupliquer les briques et optionnellement les éléments liés à cette brique.

Comme élément de départ à dupliquer on peut choisir soit :

- une page d'accueil ;
- un onglet ;

- une brique simple.

Le traitement permet de dupliquer soit :

- l'ensemble de la hiérarchie, dans ce cas chaque élément constituant la hiérarchie est lui-même dupliqué ;
- la brique de départ seule. Dans ce cas, la nouvelle brique est rattachée aux éléments déjà existants.

***Transactions concernées (ou liste des modifications)***

**TDBRQ - Traitement de duplication des briques (Transaction GTTDBQ)**

**GTBRQ - Briques (Transaction GTTBRQ)**

**GTEBQ - Eléments d'une brique (Transaction GTTEBQ)**

## **MVAR : Propagation des variantes**

Transaction permettant de dupliquer une ou plusieurs variantes d'un mnémonique modèle vers un mnémonique destination.

***Explications***

Un bouton permet de lancer la duplication des variantes sélectionnées.

Le traitement duplique à l'identique chaque variante sélectionnée sur le mnémonique destination en conservant le libellé et le défaut éventuellement modifiés dans la grille.

Une fois le traitement effectué, un message de confirmation indiquant le nombre de variantes dupliquées est affiché.

***Transactions concernées (ou liste des modifications)***

**MVAR - Propagation des variantes (Transaction GTMVAR)**

## **DUSR : Duplication d'un utilisateur**

Ajout des dates d'autorisation.

***Explications***

Actuellement les dates d'autorisation de l'utilisateur créé sont systématiquement celles de l'utilisateur modèle.

Si on le souhaite, il est désormais possible de spécifier d'autres dates d'autorisation et celles-ci seront reprises lors de la création du nouvel utilisateur.

***Transactions concernées (ou liste des modifications)***

**DUSR - Duplication d'un utilisateur (Transaction GTTDUS)**

## Modules

### Nouveautés

#### QIN - XLinks : Classes autorisées dans les directives

Dans les directives, toutes les classes java ne sont pas autorisées. Le module fonctionne sur le principe d'une liste blanche qui s'enrichit au fil du temps. A l'occasion de cette release, de nouvelles classes ont été ajoutées.

##### *Explications*

###### **java.util.ArrayList :**

Elle permet de gérer une liste ordonnée d'éléments. Les éléments s'insèrent dans la liste via la méthode "add(Object)". Ils sont accessibles via la méthode "get(int position)". Le nombre d'éléments est accessible via la méthode "size()". Cette liste peut comporter des doublons.

###### **java.util.HashSet**

Il s'agit d'un jeu non ordonné d'éléments. L'ordre d'insertion n'est pas conservé mais il n'y a pas de doublons possibles.

###### **java.util.HashMap**

Implémente une table de correspondance entre une clé et une valeur.

##### *Transactions concernées*

Sans objet

##### *Mise en place*

Cette évolution est disponible à partir de la version 181. S'agissant d'une évolution côté concepteur, aucune action spécifique n'est requise sur les interfaces.

#### QIN - XLinks : Concaténation simplifiée

Afin de faciliter les travaux de concaténation dans les directives, la variable membre "**sb**" a été mise en place.

##### *Explications*

- L'appel à "sb.clear()" vide la tampon interne.
- "sb.append(String)" ajoute un texte au tampon interne.
- "sb.toString()" renvoie le contenu du tampon interne

##### *Transactions concernées*

Sans objet

##### *Mise en place*

Disponible depuis la version 176. Pour bénéficier de cette fonctionnalité, procéder à une montée de version du moteur embarqué.

## QIN - XLinks : Fichiers dans les spools des jobs

Possibilité de générer des fichiers directement dans le **répertoire des spools** XRP grâce au nouveau type de chemin "Spool"

### Explications

Dans les directives, l'appel à la méthode "**openSpoolPatternForWrite**" de la classe "FileHandle" permet d'ouvrir un fichier en écriture directement dans les spools XRP.

Voici un exemple de code mettant en oeuvre ce type de fichiers :

```
FileHandle fhQBE = new FileHandle(this);
fhQBE.openSpoolPatternForWrite("myLog2.txt", "ISO-8859-15", true, true);
report("$1 tiers trouvés", countData("/Query/TIERS"));
report("Le fichier généré est : [$1]", fhQBE.getFile());
forEachData(data in "/Query/TIERS") {
    // sb est un objet XLinks permettant de concaténer des chaînes de caractères sb.clear();
    sb.append(val("@numoetie"));
    sb.append(";", val("@nometie"));
    sb.append(";", val("@coloetie"));
    sb.append(";", val("@dcroetie"));
    fhQBE.writeln(sb.toString());
}
fhQBE.close();
// Ajout explicite à GTCFIJ
xtjobs.addFileToGtfij(fhQBE.getFile(), true);
```

Un nouveau chemin a par ailleurs été ajouté dans les jeux de données de type fichier. Il est donc également possible d'alimenter ce répertoire directement depuis le flux de données aussi.

### Transactions concernées

Sans objet

### Mise en place

Cette évolution est disponible depuis la version 176 et nécessite une montée du moteur embarqué des interfaces.

## QIN - XLinks : Fichiers basés

Mise à disposition de la possibilité d'utiliser des **fichiers basés**.

### Explications

Jusqu'à présent, les fichiers proposés à l'utilisation dans le module étaient des fichiers présentant une existence physique sur une machine, la plupart du temps le *serveur de traitements* :

- Fichier de **type explicite**
- Fichier dans le **répertoire standard**
- Fichier décrit par paramétrage Cegid XRP Ultimate (**GPTH**)
- Fichiers générés dans les **spools** Cegid XRP Ultimate

Le module propose désormais de travailler aussi avec des **fichiers référencés et stockés directement en base de données**, à l'image de ce qui est fait avec des documents gérés en GED.

## Mise en oeuvre

Il suffit, dans les propriétés du chemin de fichier, de sélectionner le type "**Based file**", puis de renseigner le chemin souhaité. Il s'agira dans ce cas-là d'un chemin "logique" par rapport à l'arborescence basée. Les noms de lecteurs tels "C:" sont automatiquement éliminés du chemin.

## Cas pratiques

Ce type de chemin permet à une application tierce ayant accès à la base de données, mais pas au serveur de traitements, de déposer un fichier à l'attention d'une interface ou de lire un fichier issu d'une interface.

### **Transactions concernées**

Sans objet

### **Mise en place**

Disponible à partir de la version **188** du module. En génération, les options de compression et de concaténation sont pour leur part prise en compte à partir de la version **189**.

Cette fonctionnalité nécessite une montée de version du moteur embaqué de l'interface.

## **QIN - XLinks : Console d'exécution**

Mise à disposition d'un nouvel écran affichant la **liste des jobs lancés** depuis le concepteur, pour le traitement associé au programme ouvert et pour l'utilisateur connecté.

### **Explications**

Sans objet

### **Transactions concernées**

Sans objet

### **Mise en place**

Cette évolution est disponible à partir de la version **181**. S'agissant d'une évolution côté concepteur, aucune action spécifique n'est requise sur les interfaces.

## **QIN - XLinks : Verrous**

Depuis l'écran d'administration du concepteur, il est possible de **supprimer les verrous posés** sur d'autres interfaces.

### **Explications**

Sans objet

### **Transactions concernées**

Sans objet

### **Mise en place**

Cette évolution est disponible à partir de la version **181**. S'agissant d'une évolution côté concepteur, aucune action spécifique n'est requise sur les interfaces.

## QIN - XLinks : Tris

Il est possible de **sélectionner et de parcourir des données** du flux entrant depuis les directives d'une transformation. Jusqu'à cette évolution cependant, la spécification d'un **ordre de tri** n'était pas possible. Cette capacité est désormais disponible.

### Explications

Les boucles sur les données fournies par le flux entrant dans la transformation s'effectuent via l'instruction **forEachData**.

La syntaxe est la suivante :

```
forEachData(variable "XPath" [use sorters=sortersList]) {  
...  
}
```

*variable* désigne ici le nom de la variable qui, à chaque itération, prendra pour valeur la donnée courante.  
Exemple : *data*

*XPath* désigne le chemin des données à sélectionner dans l'arbre.  
Exemple : */Data/TIERS* sélectionnera tous les éléments *TIERS*, fils d'éléments *DATA* eux-mêmes présents directement sous la racine

*sortersList* désigne la liste des critères de tris à opérer sur les éléments sélectionnés  
Cette option sera exposée de façon plus détaillée dans les paragraphes suivants.

Exemple :

```
forEachData(<i>data in "/DataGroup/ITEM" use sorters=function1()) {  
  report("$1, $2, $3", val("@Id"), val("@Valeur"), val("@Code"));  
</i>
```

### Positionner des critères de tri sur une boucle

Si la prise en compte de critères de tri spécifique est souhaitée, l'option "**use sorters**" doit être ici mise en oeuvre lors de la déclaration de la boucle.

Les objets à passer à cet argument sont de type **com.qualiac.getl.akt.xml.XMLSorter**.

Exemple type :

```
// Déclaration d'un objet de tri  
boolean isCaseSensitiveG203 = false;  
XMLSorter xmlSorterG203 = new XMLSorter (SortType.ASCENDING, isCaseSensitiveG203, SortMode  
.ALPHA) {  
  public String evaluate(XMLTree tree) {  
    return tree.attrVal(null, "Id");  
  }  
}
```

};

Nous utilisons ici la déclaration en ligne d'une instance de classe abstraite. Le tri ainsi créé s'appuie sur la valeur de l'attribut "Id", utilisé au titre des expressions à comparer.

Note : pour créer de tels objets, il est possible de faire appel à la **macro "declareSorter"** du thème **"XML"** dans les méthodes proposées par le concepteur.

Pour utiliser ces objets, il suffit de les positionner dans l'instruction `use sorters` de la boucle.

Exemple : `forEachData(data in "/DataGroup/ITEM" use sorters=xmlSorterG203) {`

### ***Transactions concernées***

Sans objet

### ***Mise en place***

Cette possibilité est disponible depuis la version **213**.

Pour bénéficier de cette évolution, une montée de version du moteur embarqué est nécessaire.

## **QIN - XLinks : Extraction des interfaces basées**

Dans sa version HTML5, le module XLinks archive les "interfaces" en base de données. Il peut s'avérer utile à des fins de transfert de masse ou de sauvegarde de pouvoir **exporter ces programmes sous forme de fichiers** sur un serveur.

### ***Explications***

Le traitement permet de sélectionner les interfaces actives, éventuellement créées ou modifiées depuis une date donnée, et de les exporter sous forme de fichiers.

### **Répertoire d'export**

Le répertoire d'export est par défaut un **répertoire standard** présent sur le serveur de traitements : **\$IAC\_HOME/qetl/\_backups**. Il peut être configuré en précisant dans le critère de soumission "Exporter dans un répertoire personnalisé" la valeur N, et en positionnant un chemin pour le type **"ETB"** (créer l'occurrence "ETB" pour le paramètre TYPGPTH si ce type de chemin n'est pas disponible dans votre environnement).

### **Mode d'export**

Si le critère de soumission "Créer des instantanés" est passé à la valeur "O", le traitement créé, à chaque exécution, un sous répertoire horodaté dans le répertoire d'export et y extrait les interfaces référencées et actives détectées. De cette façon, il constitue une photo à l'instant où il est invoqué de ces interfaces.

Si ce critère est en revanche positionné à "N", le traitement extrait les interfaces directement dans le répertoire d'export. Dans le cas où un fichier d'export est présent pour une interface donnée, il ne sera écrasé que s'il présente une version antérieure à la version référencée en base.

### **Type d'export**

Les fichiers extraits peuvent être simples ou complets.

- **Fichiers simples** : seule l'interface est extraite. Cela donne lieu à des fichiers dont l'extension est "qetl". Ces fichiers sont directement exécutables par le lanceur d'interfaces. Il suffit pour se faire de les déposer dans le répertoire \$IAC\_HOME/qetl/interfaces. Attention toutefois : si l'environnement cible possède une version basée de l'interface, cette dernière sera prise en priorité par le lanceur.

- **Fichiers complets** : il s'agit d'archives dans lesquelles nous trouvons non seulement l'interface, mais également la description de ses mnémoniques, transaction et traitement. Ce format n'est pas exécutable mais peut être utilisé pour réaliser un import depuis le concepteur XLinks/HTML5. ces fichiers apparaissent avec l'extension "qetz".

### ***Transactions concernées***

**TETLBK - Sauvegarde des interfaces XLinks/HTML5 actives (*Transaction GTTETLBK*)**

### ***Mise en place***

Cette fonctionnalité est disponible depuis la version **204**.

Aucune action particulière n'est à réaliser pour bénéficier de cette fonctionnalité après passage de la release.

## **QIN - XLinks**

Mise à disposition d'un utilitaire de portage en HTML5

### ***Explications***

La ligne HTML5 amène avec elle une nouvelle version du moteur XLinks. La rupture essentielle avec la précédente (XLinks/RIA) se situe au niveau des directives, désormais codées en java.

Un utilitaire a été mis à disposition dans le concepteur XLinks/HTML5 pour assurer le portage des interfaces.

Migrer les interfaces Solstices

### ***Transactions concernées***

Sans objet

## **Modifications**

### **QIN - XLinks : Editeur de requêtes**

Correction de l'éditeur de code SQL des requêtes pour les jeux de données de type base.

### ***Explications***

La saisie par l'utilisateur d'un retour charriot dans l'éditeur de code des requêtes n'est pas prise en compte.

### ***Mise en place***

Automatique.

### **QIN - XLinks : Changer le nom de balise**

Pour les connecteurs de type délimité (CSV) ou fixes, il est désormais possible de positionner un nom de balise autre que le nom par défaut ("R").

### **Explications**

Jusqu'à présent, XLinks/HTML5 ne permettait pas de générer sous une balise autre que "R" les éléments XML issus d'une lecture de fichiers délimité ou positionnés.

Cette omission a été réparée et les propriétés spécifiques des deux types de connecteurs concernés ont été enrichies par une zone de saisie de la balise souhaitée.

### **Transactions concernées (ou liste des modifications)**

Sans objet

### **Mise en place**

Disponible depuis la version 177. Aucune action n'est nécessaire pour bénéficier de cette évolution.

## **QIN - XLinks : Sauvegarde des interfaces**

Lenteur de la sauvegarde des interfaces

### **Explications**

La sauvegarde d'une interface peut s'avérer parfois longue. Cette situation peut provenir de la présence de points de sauvegarde qui alourdissent le poids de sa description. Afin de palier ce phénomène, le stockage de ses derniers a été optimisé.

### **Transactions concernées (ou liste des modifications)**

Sans objet

### **Mise en place**

Disponible depuis la **version 184**. Aucune action n'est nécessaire pour bénéficier de cette évolution : le simple fait d'ouvrir une interface même existante met en oeuvre automatiquement le processus d'optimisation.

## **QIN - XLinks : Rechercher et remplacer**

Evolution de la fonctionnalité de recherche du concepteur :

- Suite à la recherche, l'arbre de recherche est déplié automatiquement.
- Possibilité de passer à la recherche suivante ou précédente (boucle sur les recherches).
- Ajout des fonctionnalités pour remplacer le texte sélectionné dans le code et pour remplacer toutes les occurrences trouvées lors de la recherche.

### **Explications**

Sans objet

### **Transactions concernées**

Sans objet

### **Mise en place**

Cette évolution est disponible à partir de la version 181. S'agissant d'une évolution côté concepteur, aucune action spécifique n'est requise sur les interfaces.

## **RGPD : Possibilité de gérer l'anonymisation pour une adresse de tiers, sans traiter sur le tiers**

Evolution de la fonctionnalité de recherche et de traitement des données personnelles.

### **Explications**

Lors de la recherche et du traitement des informations personnelles il est possible de définir que l'on ne remonte jamais sur la fiche tiers. Il est ainsi possible de traiter l'anonymisation des informations personnelles en limitant les recherches et en contrôlant les validités pour le couple adresse/tiers.

### **Transactions concernées**

**GKRGPD - Contrôle des données personnelles (Transaction GTCRGPD)**

**GTRGO - Contexte des données personnelles (Transaction GTIRGO)**

**ERGPD - Edition des données personnelles (Transaction GTERGPD)**

**ERGPDE - Extraction des données personnelles (Transaction GTERGPDE)**

**TRGPD - Anonymisation (Transaction GTTRGPD)**

**TLRGPD - Anonymisation par liste (Transaction GTLRGPD)**

**TMRGPD - Trt. de modification des données personnelles (Transaction GTMRGPD)**

### **Mise en place**

Afin de ne plus traiter la fiche tiers, et ce limiter aux adresses de tiers, il faut :

- Rendre inactif les enregistrements du contexte QPOETIE dans la gestion des contextes des données personnelles.
- Rendre inactif l'occurrence QPOETIE du paramètre NUMGTRGO.
- Rendre actif les enregistrements du contexte QPOETIA dans la gestion des contextes des données personnelles.
- Rendre actif l'occurrence QPOETIA du paramètre NUMGTRGO.

## **QIN - XLinks**

Le gestionnaire d'interfaces fait désormais apparaître le **nom du mnémonique** par défaut associé à l'interface.

### **Explications**

L'écran du gestionnaire d'interfaces fait apparaître un certain nombre d'informations jugées importantes. Le mnémonique par défaut associé aux interfaces présentées est désormais également proposé.

### **Mise en place**

Disponible depuis la **version 216**

## **QIN - XLinks**

Corrections mineures diverses.

### **Explications**

#### **Proposition de code.**

La séquence de touches {Ctrl + Espace} invoque la fonctionnalité de proposition de code. Il s'agit ici de présenter à l'utilisateur des propositions de variables, méthodes ou classes qui pourraient faire sens dans le contexte où il est positionné.

Il s'avère que cette proposition restait muette dans le cas des variables internes aux boucles formulées par l'instruction "forEach". Ce point a été corrigé (anomalie 1487).

***Transactions concernées (ou liste des modifications)***

Sans objet.

***Mise en place***

Automatique.