

# Mia-Insight Serveur Release note 5.2.0

## Table des matières

API D'INTERROGATION DU REFERENTIEL.....	2
EVOLUTION DU PLUGIN JENKINS POUR MIA-TASKMANAGER .....	3
PERFORMANCES .....	3
DEPLOIEMENT .....	3
DETAILS DE LA VERSION .....	3
1.1.1 Serveur Mia-Discovery v5.2.0 .....	3
1.1.2 Plugin Jenkins pour Mia-TaskManager v2.2.0 .....	5
1.1.3 Parser Cobol v7.04.16 .....	5
1.1.4 Parser DotNet v3.1.15 .....	5
1.1.5 Parser HTML/JS v4.1.2 .....	9
1.1.6 Parser Java v5.0.12.....	10
1.1.7 Parser JCL v3.7.2.....	10
1.1.8 Parser PLSQL v3.0.4 .....	11
1.1.9 Parser SQL v6.0.29.....	11
1.1.10 Parser VBNet v3.0.4 .....	15
1.1.11 Parser XML v2.0.4.....	15

Mia-Software a le plaisir de vous informer que la **nouvelle version serveur 5.2.0 de Mia-Insight** est disponible.

Cette version majeure apporte de nouvelles fonctionnalités ainsi qu'un grand nombre d'évolutions et de corrections dont vous trouverez la liste ci-après.

## API D'INTERROGATION DU REFERENTIEL

La version 5.2 expose désormais une API RESTful publique et documentée permettant d'interroger directement le référentiel sans utiliser le plugin pour Eclipse. Il est ainsi possible de bénéficier de toute la puissance de Mia-Discovery avec une simple url.

Les applications sont multiples : interrogation en mode batch, exécution de requêtes directement à partir d'un portail web, recherches textuelles invoquées depuis une interface de développement ...

**Mia-Discovery Server API** 1.0.0  
[ Base URL: mia-dev.sodifrance.fr ]  
[https://mia-dev.sodifrance.fr/mia-discovery/multitechno/5.2.0/api/doc/swagger/json?group=Mia-Discovery Server API](https://mia-dev.sodifrance.fr/mia-discovery/multitechno/5.2.0/api/doc/swagger/json?group=Mia-Discovery%20Server%20API)

Définition de toutes les API publiques offertes par un serveur Mia-Discovery, regroupées par version ou par service. Chaque API est accompagnée d'une description, des différents paramètres attendus et du format de la réponse. Des templates d'utilisation dans différentes technologies sont consultables sur la page des [usages](#).

Pour faciliter les développements :

- Définition OpenAPI au format yml
- Définition Swagger au format json

Pour plus d'informations :

- Specification officielle OpenAPI > <https://github.com/OAI/OpenAPI-Specification/>
- Définition officielle des codes retour HTTP > <https://tools.ietf.org/html/rfc2616#section-10>

Terms of service  
Contact the developer  
Mia-Software Software User Agreement

**authentication** Regroupe l'ensemble des fonctions de l'API gérant l'authentification.

**objects** Regroupe l'ensemble des fonctions de l'API permettant d'accéder aux objets du référentiel.

**sources** Regroupe l'ensemble des fonctions de l'API permettant d'accéder aux sources du référentiel.

**queries** Regroupe l'ensemble des fonctions de l'API permettant d'exécuter des requêtes sur les objets du référentiel.

**metamodel** Regroupe l'ensemble des fonctions de l'API permettant d'accéder au métamodèle du référentiel.

Les principales fonctions proposées dans cette version sont les suivantes :

- Recherche de composants du référentiel
- Recherche des requêtes disponibles et applicables sur une population
- Exécution de requêtes sur une population de composants
- Recherche textuelle sur les sources indexés

Les résultats sont retournés au format JSON.

La documentation, des exemples et la possibilité de tester directement l'api sont disponibles à l'adresse suivante [http\(s\)://<server>:<port>/api](http(s)://<server>:<port>/api).

**objects** Regroupe l'ensemble des fonctions de l'API permettant d'accéder aux objets du référentiel.

GET [/api/objects/byName](#) Recherche des objets du référentiel par leur nom.

GET [/api/objects/byOid](#) Recherche des objets du référentiel par leur identifiant.

GET [/api/objects/byQuery](#) Exécute une requête sur les objets du référentiel.

Exécute une requête pour les objets sources transmis en paramètre, et retourne les n-uplets d'objets résultats.

Name	Description
mapping	Boolean indiquant que le nom mappé des attributs et relations est retourné si valorisé à true. L'id technique est retourné si valorisé à false.
oids * required	Liste d'identifiants d'objets séparés par une virgule.
pageNumber	"Numéro de page souhaitée ; la numérotation commence à 1. La propriété 'totalPages' de la réponse fournit le nombre total de pages, et la propriété 'links' les urls vers les autres pages. Note : la valeur particulière 0 peut être utilisée pour obtenir la totalité des éléments sans pagination."

Default value : true

Default value : 1

Server response

Code	Details
200	Response body

```

{
  "rows": [
    {
      "objects": [
        {
          "oid": "HostSoftCompSource",
          "entity": "Programme",
          "lowerUserId": 0,
          "attributes": [
            {
              "name": "Nom",
              "type": "String",
              "value": "RVSPW01W"
            }
          ]
        },
        {
          "name": "Type",
          "type": "String",
          "value": ""
        }
      ]
    },
    {
      "relations": [
        {
          "oid": "_143541c6029da8_45883",
          "entity": "Accesneur de données",
          "lowerUserId": 0,
          "attributes": [
            {
              "name": "Mode"
            }
          ]
        }
      ]
    }
  ]
}

```

Après une période d'essai de 3 mois, cette fonctionnalité sera soumise à licence venant en supplément de votre configuration actuelle.

## EVOLUTION DU PLUGIN JENKINS POUR MIA-TASKMANAGER

Afin d'apporter encore plus de souplesse dans l'ordonnancement des chaînes analyses, il est désormais possible de lancer plusieurs chaînes au travers d'un seul job Jenkins et obtenir leurs différents arbres et rapports d'exécution.

Retour au projet

État

Modifications

Console Output

View as plain text

Informations de la construction

Supprimer le build

Git Build Data

No Tags

Voir les empreintes numériques

Build précédent

Build suivant

### Construction #26 (5 nov. 2020 20:27:57)

Exécution de l'EPR : Demo

- ▼ Demo [00:55:45]
  - ▶ Préparation [00:02:00]
  - ▶ Analyses paralleles [00:01:43]
  - ▶ Analyse des ordres SQL [00:00:07]
  - ▶ CHARGEMENT MIA-DISCOVERY [00:51:54]
  - Envoi code retour [00:00:00]

Exécution de l'EPR : Demo - Post vérifications

- ▼ Demo - Post vérifications [00:00:00]
  - ▶ Non régression - Comparaisons [00:00:00]
  - Envoi code retour [00:00:00]

## PERFORMANCES

Un gros travail d'optimisation de la phase de chargement a été effectué permettant ainsi de diviser par 5 les temps de chargements des fichiers de résultats issus des analyseurs Java et DotNet. Au final, et suivant la proportion de ces langages dans vos patrimoines, il est possible de gagner de 1 à 5 heures sur vos temps de traitement.

## DEPLOIEMENT

Se reporter aux documents « Mia-Insight-5.2.0\_Architecture\_et\_prerequis.pdf », « Mia-Insight-5.2.0\_Administration.pdf » et « Mia-Insight-5.2.0\_Protocole de montée de version.pdf » pour les modalités de déploiement de cette version.

## DETAILS DE LA VERSION

Cette version apporte précisément les améliorations qui vous sont présentées ci-dessous :

### 1.1.1 Serveur Mia-Discovery v5.2.0

- v5.2.0
  - Nouveautés

- L'authentification supporte à présent le "Single Sign On" ([SSO](#)) via le protocole [SAML](#). Pour plus d'information se reporter à la documentation d'administration.
- Le serveur Mia-Discovery expose à présent une [RESTful API](#) basée sur [OpenAPI](#) permettant d'interroger le référentiel. La documentation de cette API est disponible sur votre serveur à l'adresse : `http(s)://<server>:<port>/api`
- Le nombre de lignes indexées par Elasticsearch est maintenant retourné au client Mia-Discovery. {MIA-333}
- Amélioration significative des temps de chargement, notamment pour les patrimoines Java et DotNet.
- Chargement
  - Ajout dans l'imp standard de :  
`LOAD *.PGM_TXN_CALL`
  - Consolidation des JCL, gestion de la clause ALL dans les LISTCAT. {5564}
  - Correction du chargeur SC\_OBJ\_SC\_OBJ\_PLUS qui ne gérait pas correctement les objets non prévus en entrée. {5758}
  - Au `LOAD *.N_OBJ_ATT` si une valeur d'un attribut est invalide (ex : incompatible avec le typage, en dehors de la plage des valeurs possibles) alors une erreur de chargement apparaît à présent dans le rapport.
  - Amélioration de l'utilitaire "RepositoryTypesCountComparison.bat" : à présent pour la comparaison des statistiques à la fin du chargement, pour les types la clé utilisée est l'ID de l'entité et pour les relations le RELATIONID, afin que la comparaison ne soit plus sensible aux variations sur le mapping. {5468}
  - Amélioration des temps d'exécution des consoliders des assignations {5623}
  - Optimisation des temps de chargement pour les chargeurs CHARACTERISTIC\_DEFINEDOBJECTTYPE et JAVACLASS\_LAUNCHEDMETHOD.
  - Correction du chargeur NAV\_ACTION\_USE\_METHOD : création de la méthode pointée si elle n'existe pas (création précisée dans le rapport de chargement). {3202}
  - Optimisation des temps de chargement pour le chargeur RTN\_DCL\_SPT. {5632}
  - Les chargeurs "TRG\_INFO\_SUP", "REL\_VIEW", "REL\_INDEX\_REL\_TABLE", "REL\_FOREIGN\_KEY\_REL\_TABLE", "REL\_KEY\_MULTI\_COLUMNS" et "REL\_INDEX\_MULTI\_COLUMNS" sont compatibles avec l'option "multiple.find". {MIA-664}
  - Les chargeurs "CHARACTERISTIC\_DEFINEDOBJECTTYPE" et "OBJECTTYPE\_DEFINEDJAVASOURCE" créent maintenant correctement les Signets et objets de documentation. {5755}
- Corrections diverses
  - Correction d'erreur dans les flux envoyés au client Eclipse pour les valeurs d'attributs de type : double, float, byte.

- Correction erreur bloquante en cas d'erreur de syntaxe d'un lien d'une icône de viewpoint. {5361}
- Correction du consolideur JAVAOVERRIDINGMETHOD qui ne s'arrêtait jamais dans certains cas complexes.
- Correction des fichiers lanceur Mia-Task Manager pour le chargement Elastic (mise à jour des descriptions).
- Correction dans l'exécution de requêtes ayant une condition sur un élément non visible {5793}

### 1.1.2 Plugin Jenkins pour Mia-TaskManager v2.2.0

- v2.2.0 (09/2020)
  - Affichage de plusieurs arbres en cas de plusieurs exécutions d'EPR dans un même job.
  - Correction de l'affichage du rapport des étapes avec caractères spéciaux dans leur nom.
  - Correction de la coloration des étapes.
  - Correction de la prise en compte des paramètres dans la configuration des jobs Jenkins.
  - Correction de la récupération des rapports contenant des caractères '\*', on obtenait un résultat tronqué dans l'arbre d'exécution.
  - Correction dans la gestion des espaces dans la transmission de paramètres à l'exécution de l'EPR.
  - Correction dans l'affichage de l'arbre une étape pouvait être affichée en rouge "erreur" alors qu'aucun de ses enfants n'était en erreur.

### 1.1.3 Parser Cobol v7.04.16

- 08/10/2020 (7.04.16)
  - Gestion d'un cas particulier de ligne dans clause remarks avec un mot == à ne pas confondre comme un début de pseudo-texte
  - Amélioration du rapport :
    - Ajout de lignes d'informations (ligne de commande, version de l'outil, début, fin...)
    - Formatage spécifique lorsque le fichier de rapport comporte l'extension \*.log

### 1.1.4 Parser DotNet v3.1.15

- v3.1.16 (30/10/2020)
  - projets :
    - correction cas homonymes avec impact sur rlc Mia-Discovery absence vidage PROJECT\_SOURCE + ajout d'un pathId en sorties PROJECT\*
    - prise en compte des projets website par les fichiers .sln
- v3.1.15 (22/10/2020)
  - xrefs de types :
    - correction résolution vers des innerClassifiers (a nécessité gestion partial class et recherche en cascade)

- types génériques : prise en compte du cas où le préfixe est aussi un type générique
  - rlc Mia-Discovery : sortie CHARACTERISTIC\_DEFINEDOBJECTTYPE rectification de certaines propriétés (extern, ...) lié à correction fonction HasTaggedValue qui testait sur name au lien dataValue
- v3.1.14 (09/10/2020)
  - rlc Mia-Discovery :
    - sortie CHARACTERISTIC\_USEDMETHOD :
      - ⇒ ne plus lister appel constructeur non trouvé par new (déjà listé dans JAVATYPE\_USEDMETHOD)
      - ⇒ suppression des paramètres de type générique dans les références UNKNOWN (l'ObjectId généré était erroné)
    - ObjectId des accesseurs de property et index : refonte pour mise en commun et ajout pathId
  - rlc SQL :
    - ajout sortie lineNumber
    - cas argument :
      - ⇒ restituer ToStringSQL au lieu Value (supprime notamment les quotes englobantes)
      - ⇒ ajout du cas de concaténation opérateur Plus
    - ne plus découper les multi-instructions (séparateur ;)
  - consolidation types fin passe 1 :
    - recentraliser les types non résolus dans package SYSTEM (but : meilleur exploitation dans les sorties xmi)
    - respect de l'ordre dans la relation Classifier.parentClassifier (impact notamment pour instruction base en xrefs)
  - résolutions xrefs types : recherche des innerClassifiers dans le source et abandon recherche de secours dans packages racines
  - rlc LOCALVARTYPEINFERENCE : correction/amélioration déduction sur new et cast
  - littéraux template (TemplateLiteral) : amélioration restitution ToStringSQL() et Evaluate()
- v3.1.13 (02/10/2020)
  - xrefs :
    - resolution using aliassé : correction
    - types génériques : correction pour prendre en compte l'éventuel préfixe
    - types homonymes : les types recherchés ne sont plus stockés dans SYSTEM, mais localisés dans le SourceFile courant (relation owns)
      - + de même pour les types génériques et types tableaux (auparavant dans ORPHANTYPE)
    - FindOrCreateType : ne plus forcer name si notation pointée (notamment à cause using aliassé)
  - écrémage des faux types génériques (liés aux déclarations de methodes paramétrées et leurs appels)
  - rlc Mia-Discovery : en lien avec la relocalisation des types recherchés, génériques et tableaux, production de ces items au fur et à mesure de chaque source plutôt qu'en fin de traitement

- correction plantage potentiel (et avéré suite évolutions) sur cas de liste avec CreateString (accès au lineNumber)
- rapport : ajout du message d'information de Consolidation en cours
- v3.1.12 (25/09/2020)
  - correction d'un problème de recherche de type Classifier en passe 1 retournant un objet Method (qui mettait le source en erreur avec message rapport "ERROR: Element not found !!!")
  - grm :
    - ajout cas de tableau de tuple
    - nom de champs de tuples implicites
    - syntaxe <, > ajout prise en compte plusieurs virgules
    - ajout cas sur null condition operator ?[ suivi de test is ...
    - généralisation mot-clé default de code générique
  - scanner : template literal amélioration lecture portions textuelles
  - purge des corps de méthode passe 2 : correction pour éviter si méthode locale cause plantage (iterateur relation Defines)
  - rlc METHOD VARLOCAL et SQL : exclure le traitement du package default au final, provoquant plantage car cast en SourceFile donc accès à données fantomes
  - rlc MDDScript : correction sur un message d'erreur sur mauvais iterateur
- v3.1.11 (18/09/2020)
  - correction bouclage sur Evaluate (notamment string.Format)
  - correction xrefs sur recherche de namespaces + optimisation recherche xref pointée
- v3.1.10 (17/09/2020)
  - fichiers projet csproj : amélioration détection .net Core Sdk
  - optimisation xrefs sur recherche de namespaces (effet de bord : corrige quelques cas de xrefs)
  - fonction Evaluate() : syntaxe string.Format / string.Concat prise en compte
- v3.1.9 (08/09/2020)
  - grammaire / scanner :
    - correction sur null condition operator ?.
    - ajout cas sur null condition operator ?[ suivi de test !=
    - correction nullableInterrogationPoint (cas de cast as xxx? suivi de :)
  - fichiers projet csproj : prise en compte des spécificités .net Core Sdk, dont les sources dans l'arborescence (BuildFileList)
  - MDDScript : ajout fonction OperationCall.GetArgument(int n) : Argument
- v3.1.8 (03/09/2020)
  - résolution xrefs :
    - recherche, si dans package SYSTEM -> durcissement de la restitution par défaut (cf Package.Find, nombre de TypeParameter)
    - déduction type instancié : prendre en compte le cas où il y a plusieurs niveaux avant de trouver la classe template

- rlc Mia-Discovery : ne pas produire dans sortie PROJECT\_EXE la ligne si assembly vide
- v3.1.7 (01/09/2020)
  - MDDScript :
    - fonction Classifier.FindMethod ne fonctionnait pas
    - ajout fonction Classifier.FindFeature(RuntimeClass, string) : Feature
    - ajout fonction Classifier.IsOfType(string) : bool
  - résolution xrefs : ajout gestion des appels de méthodes templatés et classes parentes génériques instanciées pour déduction type instancié au lieu du type paramétré générique
- v3.1.6 (25/08/2020)
  - optimisation delete final (suppression pck SYSTEM en premier ainsi que l'éventuel pck System (dans le cadre de l'option -s)
  - correction plantage delete final en cas de source asp.net (lié à ajout pck système ORPHANTYPE)
  - scanner :
    - correction lecture des éléments antislashés dans les string interpolation (\; bloquait notamment sur double \\)
    - ajout d'un cas de async\_lambda
  - passe 1 :
    - delegate ajout traitement des parents
    - corrections sur destruction d'objet cas framework interne (avec le bon argN)
    - methodes d'interface : modélisation des TypeParameter + correction gestion modifiers (stereotype "NEW" -> isNew)
  - passe 2 : ajout cas UnaryOperator en qualifier via CSHARPARithmeticalExpression (notamment pour await; GenerateExpression; perdue auparavant)
  - algo xref Find : élimination codage double recherche avec puis sans nb params au profit d'un resultat par défaut
  - évolution recherche FindMethod pour tenir compte du nombre de TypeParameter lors de l'OperationCall
  - correction rattachement des TypeParameter des Method inconnues
  - correction lien TypeParameter.value (inverse valued) qui pouvait disparaître lors de consolidation des types fin passe 1
  - modélisation des OperationCall : ajout des TypeParameter
  - modélisation des TypeParameter : ceux pour les Templates pointent en value sur un Classifier local qui centralise les éventuelles clauses where (via parentClassifier), ainsi que les utilisations
  - évolution du MM ASPNet :
    - ajout lien OperationCall.firstTypeParameter vers CSHARPTypeParameter => nouveau plugin EMF com.mia.itmf.aspnet.metamodel v2.6.0
- v3.1.5 (30/07/2020)
  - types paramétrés :

- amélioration résolution xrefs via le nb de params
  - refonte gestion double recherche avec puis sans nb params (corrige aussi un pb dans algo recherche qui faisait un return sans tester NULL et qui shuntait donc la suite)
- types génériques GenericTypeInstanciation : ajout résolution xrefs autour des Find
- v3.1.4 (27/07/2020)
  - mantis 1459 : amélioration restitution types tableaux
    - nommage ARRAY OF type devient type[] (restitution des [],,)
    - rlc Mia-Discovery :
      - ⇒ LUTID 0 au lieu 0x11
      - ⇒ référencement dans sortie OBJ N\_OBJ + lien vers type element via nouvelle sortie JAVATYPE\_TYPEDARRAYTYPE\_XREF
  - types framework : l'alias bool n'était pas mappé sur System.Boolean (c'était boolean) + optimisation algos recherche
  - parsing fichier projet csproj : ne pas restituer la version dans la dépendance de projet (Reference Include = " ... , Version=...")
  - grm : suppression distinction Array\_Creation\_Exp/Non\_Array\_Creation\_Exp qui n'était plus utile et empêchait le cas new uint[] { ... }[ x ]
  - algos recherche dans Find avec Classifier.innerClassifiers et FindMethod avec Classifier.features passent en case sensitive (respect casse)
- v3.1.3 (30/06/2020)
  - scanner/parser C# 6 : string interpolation améliorations capture et représentation
  - passe 2 : correction bug création à tort TemplateLiteral au lieu StringLiteral si chaîne "\$..." (depuis v3.1.0)
  - évolution du MM ASPNet : ajout TemplateLiteralFormattedExpr pour les expressions formatées dans un template literal (alignment, format)  
=> nouveau plugin EMF com.mia.itmf.aspnet.metamodel v2.5.0
- v3.1.2 (25/06/2020)
  - rlc Mia-Discovery : ajout pour chaque projet en entrée du fichier projet en documentation (N\_OBJ\_DOC)
  - harmonisation des rapports :
    - ajout commande, version, timestamp début/fin
    - nouveau format log (activé par présence .log dans le nom du rapport)

### 1.1.5 Parser HTML/JS v4.1.2

- v4.1.2 (11/09/2020)
  - grammaire :
    - cas de fonctionExpression async anonyme
    - objectDefinition gérer cas de points-virgules multiples (empty ';')
    - literalField cas de fonctions getter/setter (set)

- sorties res : les xrefs vers fonctions sont systématiquement avec chemin du source
- sources .js/.jsx en erreur de syntaxe -> le niveau passe de Severe Warning à Error
- amélioration performances :
  - index des sources par nom dans le projet (model)
  - correction fonctions internes Get\_rev\_field et Get\_rev\_leftHandSide qui pouvaient renvoyer un élément erroné
- résolution imports/exports : correction sur import en mode projet de source vers mauvais homonyme (respect casse)
- v4.1.1 (30/03/2020)
  - mantis 5371 : correction plantage sur new de type tableau / mise à niveau javascript (DotNetParser)
- v4.1.0 (27/01/2020)
  - passage à 2 passes avec consolidation des imports/exports
  - amélioration résolution déduction types sur new (par report en conso fin passe 1 (ou fin de source hors parseur HTMLJS), où lien elements alimenté)
  - évolution du MM pour les ImportDeclaration / ExportDeclaration et héritage ASTNode au lieu Statement + utilisation Modifier avec ajout attribut isDefault => nouveau plugin EMF com.mia.itmf.htmljs.metamodel\_2.4.0
  - modification résolution des imports + exports
  - correction : les xrefs résolues via import n'étaient pas restituées dans le lien GetXrefs
  - résolutions : bascule en case sensitive ( + impact potentiel en scripting sur les fonctions FindXXX)
  - rlc Mia-Discovery : sortie \*USE\_TYPE : il manquait les manipulations simples (instanceof, ...)

### 1.1.6 Parser Java v5.0.12

- v5.0.12 (15/09/2020)
  - mantis 5756 : correction plantage sur import pck.Typ.\* lors rlc Mia-Discovery si pck.Typ déjà référencé (mais source non fourni; ex : java.util.List.\*)
  - résolution xref simplification interne cas .class (avec amélioration)
  - correction fonctions internes Get\_rev\_field et Get\_rev\_leftHandSide qui pouvaient renvoyer un élément erroné (concerne notamment la fonction Evaluate) (amélioration possible des performances en passe 2)
- v5.0.11 (27/08/2020)
  - mantis 5710 : éviter stack overflow (taille de la pile) en lecture commentaires monolignes consécutifs

### 1.1.7 Parser JCL v3.7.2

- v3.7.2 (23/07/2020)
  - Fix : Certaines cartes inlines d'une PROC n'étaient pas correctement affectées à la carte DD \* correspondante

- v3.7.1 (22/07/2020)
  - Fix : Gestion des espaces dans les ordres RUN PROGRAM() pour IKJEFT01

### 1.1.8 Parser PLSQL v3.0.4

- v3.0.4 (23/10/2020)
  - rlc Mia-Discovery et OPENCURSOR, sorties SC\_OBJ\_REL\_ENT\_CS : manquait la colonne objectId 0XD2000000;UNKNOWN; qui empêchait la bonne prise en compte du n° de ligne par le sqlparser
- v3.0.3 (04/09/2020)
  - grammaire :
    - généralisation expression unaire NOT (notamment pour cas NOT xxx NOT IN xxx qui n'était pas pris en compte [mantis 5630])
    - ajout cas de commande PROMPT
    - ajout commande DROP du cas TYPE
  - correction plantage sur script si déclaré après routine de même nom que le fichier source -> forcer un suffixe pour le script
- v3.0.2 (20/07/2020)
  - mantis 5630 : prise en compte CREATE TRIGGER COMPOUND TRIGGER

### 1.1.9 Parser SQL v6.0.29

- 6.0.29 (02/11/2020)
  - mantis 5827 : ajout statement DB2 SET PATH [=] xx, yy ...
  - mantis 5828 : autoriser caractères accentués en première lettre d'un identifiant
- 6.0.28 (30/10/2020)
  - mantis 5826 : correction plantage sur DECLARE sans variable locale
  - sécurisation sur détection token CURSOR (m\_pCurrentStatementIdentifier)
  - correction INSERT ALL token de début de statement (m\_pCurrentStatementIdentifier)
- 6.0.27 (23/10/2020)
  - mantis 5630 fin :
    - cas de IF:NEW pris en label; récupération dans le scanner
    - UPDATE cas de RETURN xxx INTO yyy en PL/SQL
    - INSERT cas de VALUES index de tableau de row (style fonction\_call)
- 6.0.26 (15/10/2020)
  - mantis 5771 suite :
    - valeur par défaut de paramètre DEFAULT USER
    - valeur initiale de variable locale :
      - ⇒ à NULL sans signe ':='
      - ⇒ élargissement à expression (cas de comparaison xxx = yyy)
    - quoted string literal en mode oracle (q'[...]' )
    - ALTER USER QUOTA UNLIMITED ON
    - GRANT ajout privilège CREATE ANY SYNONYM
    - CONNECT BY / START WITH :
      - ⇒ ajout parenthésage et opérateur OR dans la liste de clauses

- ⇒ CONNECT BY NOCYCLE
- ⇒ refonte pour permettre cas CONNECT BY standalone
- ORDER BY : ajout des cas -/+ expr unaire
- gestion opérateur MOD du PL/SQL Oracle
- SELECT INTO target ajout cas ligne de record (ex : into PACK\_AUTO.TABLE\_FLOTTD(1))
- clause PIVOT fonction AS name
- predicat xxx IN function\_call
- clause EXISTS cas de double parenthésage
- fonction EXTRACT forme standard function\_call
- instruction CONTINUE WHEN condition
- EXECUTE IMMEDIATE clause RETURNING INTO
- query\_exp\_nopar ajout cas suite union parenthésée
- argument de function\_call ajout cas de comparaison
- ajout instruction INSERT ALL
- remodelage function\_call/tableRecordAccess (pour doubles calls) + couvre cas de instruction sur donnée .DELETE(n)
- ajout instruction sur donnée .EXTEND
- CASE WHEN cas de test avec booléen AND test
- gestion USE en tant qu'identifiant (reconnaissance du cas d'utilisation en statement)
- paramètres de function\_call : possibilité de AS name (utilisation de scalar\_exp\_or\_NULL)
- ajout gestion fonction XMLSERIALIZE dans le cas syntaxe Oracle
- v6.0.25 (24/09/2020)
  - mantis 5770 : autoriser la commande SHOW en mode -oracle (évite d'utiliser -sqlPlus réservé pour les scripts)
  - mantis 5771 partiel :
    - CREATE FUNCTION/PROCEDURE/TRIGGER après PROMPT sans séparateur
    - types de FUNCTION ajout DETERMINISTIC, PARALLEL\_ENABLE, RESULT\_CACHE et AGGREGATE USING
    - SHOW ERRORS PACKAGE [BODY] xxx
    - CONNECT BY et START WITH : extension de comparison\_predicate à predicate
    - type NUMBER avec float (ex : NUMBER(10.0))
    - correction CREATE TYPE BODY
    - requêtes SELECT ajout clause INTO
    - fonctionnalité OVER avec clause ORDER BY : ajout cas de clause PRECEDING
    - END LOOP ajout cas avec id label
    - opérateur CONNECT\_BY\_ROOT
- v6.0.24 (17/09/2020)
  - mantis 5762 : prise en compte de l'encodage pour les entrées fichiers résultats de parseurs (mode non script (pas -s); ouverture fichier en mode bin + positionnement selon -charsetSrc)

- v6.0.23 (10/09/2020)
  - scanner : tokens id ajout prise en compte 2 caractères spéciaux
  - grm :
    - instruction SET avec DEFERRED/IMMEDIATE
    - host numérique
    - notation (+) sur host
- v6.0.22 (04/09/2020)
  - correction gestion clause contrainte USING INDEX qui générait à tort une table de ce nom
- v6.0.21 (28/08/2020)
  - mantis 5695 :
    - ALTER SESSION SET EVENTS
    - CREATE USER IDENTIFIED BY VALUES + id ou string
    - COMMENT ON MATERIALIZED VIEW
    - ALTER INDEX NOPARALLEL
    - GRANT privilege QUERY REWRITE / DEBUG / FLASHBACK en 1er élément
    - ALTER TRIGGER COMPILE
    - BEGIN plsql END id ;
    - refactoring ADD SUPPLEMENTAL LOG DATA
    - CREATE TABLE
      - ⇒ cas OF XMLTYPE
      - ⇒ clauses :
        - modification ORGANIZATION EXTERNAL et avec token ':'  
+ clauses REJECT LIMIT UNLIMITED/number et PARALLEL  
number optionnel
        - ajout cas LOB store BASICFILE avec nom string
        - ajout clauses XMLTYPE STORE AS, XMLSCHEMA et OID
    - CREATE BITMAP INDEX
      - ⇒ function\_call :
      - ⇒ ajout cas string.xxx() avec impact sur string.xxx.yyy et  
column\_ref\_extend
      - ⇒ ajout cas trigger :NEW.xx.yy() avec impact sur ArgExpression
    - suppression des lignes vides dans la console
- v6.0.20 (20/07/2020)
  - mantis 5630 & DDL même client : grm
  - CREATE OR REPLACE EDITIONABLE PACKAGE
  - ALTER TRIGGER ENABLE/DISABLE
  - CREATE OR REPLACE EDITIONABLE TRIGGER
  - refonte CREATE FORCE VIEW cause conflit EDITIONABLE
  - instruction FOR LOOP END LOOP label
  - instruction EXIT id WHEN condition
  - ajout PROCEDURE locale
  - ALTER TABLE ajout clause ADD SUPPLEMENTAL LOG GROUP

- ALTER TABLE clause ADD SUPPLEMENTAL DATA ajout cas UNIQUE INDEX
  - valeur DEFAULT de colonne ajout cas NEXTVAL de sequence
  - ajout system variable SYSTIMESTAMP
  - ajout cas DEFAULT SYS\_CONTEXT(...) (nb : dans scanner)
  - ajout trigger composé CREATE TRIGGER .. FOR .. COMPOUND TRIGGER ...
  - CREATE OR REPLACE TRIGGER accessMode ajout INSERT/DELETE OF list
  - INTERVAL mot-clé contextuel <INTERVAL\_FUNC>
  - FETCH cursor INTO ajout cas arguments préfixés '!'
- v6.0.19 (16/07/2020)
    - mantis 5624 : CREATE TABLESPACE ajout clause PAGENUM
  - v6.0.18 (07/07/2020)
    - mantis 5591 : mode -genCSFile sortie TRG\_INFO\_SUP correction colonne type evenement (int 2 -> int 1)
    - mantis 1968 :
      - correction FOREIGN KEY lien table - colonnes de FK
      - amélioration au mantis 5471 : gestion UNION dans les query expression select. (nb : reste à gérer découpage CREATE VIEW)
  - v6.0.17 (22/06/2020)
    - grm : corrections :
      - gestion permissivité virgule fin sélection et set de update
      - ajout prise en compte fonction XMLPARSE
      - contextualisation contraintes mySql INDEX gênantes sur autre SGBDR en cas de nom de colonne index
      - scanner : pb typage en fin d'entrée des mots-clés fonction (identifier dans ce cas; principalement en mode non script)
    - harmonisation des rapports :
      - ajout commande, version, timestamp début/fin
      - nouveau format log (activé par présence .log dans le nom du rapport)
  - v6.0.16 (12/06/2020)
    - mantis 5471 : rendre étanche les UNION des statements SELECT, notamment pour cohérence déduction du lien table-colonne (cf code v5.3.0 11/05/11 )
  - v6.0.15 (05/06/2020)
    - grm : ajout cas d'opérateur concat '||' (166)
  - v6.0.14 (28/05/2020)
    - CREATE TRIGGER ajout :
      - cas NO CASCADE BEFORE
      - cas FOR EACH STATEMENT
      - options SECURED / NOT SECURED
      - block plsql ajout cas instructions CALL et SET x.y (ex : SET x.STATUS\_DATE = CURRENT\_TIMESTAMP)

- grm des expressions ajout cas CURRENT TIMESTAMP WITH TIMEZONE ayant nécessité :
  - correction v6.0.2 WITH TIME ZONE (le <WITH\_TIME> n'était pas géré)
  - correction keyword fonction TIMESTAMP  
+ ajout potentiel cas CURRENT TIMESTAMP ( x )
- sortie N\_OBJ : ajout en description de l'instruction du CREATE INDEX pour les INDEX
- v6.0.13 (18/05/2020)
  - mantis client 5763 : DB2 CREATE TABLESPACE option INSERT ALGORITHM
  - mantis 5418 : CREATE TABLE liste des colonnes cas de doubles quotes
- v6.0.12 (02/03/2020)
  - mantis 5348 grm oracle :
    - PARTITION BY RANGE INTERVAL + PARTITION identifiants string possibles
    - FOREIGN KEY actions intégrées aux contraintes states cause position intercalée possible (ON DELETE CASCADE ENABLE)
    - column def clause IDENTITY :
      - ⇒ ajout cas GENERATE ON NULL
      - ⇒ réagencement et ajout cas create\_sequence\_option\_lst non parenthésé (nécessitant remaniement de cette clause)
    - USING INDEX cas de LOCAL PARTITION et CREATE INDEX à la volée
    - LOCAL PARTITION identifiants string possibles

#### 1.1.10 Parser VBNet v3.0.4

- v3.0.3 (13/10/2020)
  - mise à niveau et report de corrections du DotNetParser  
=> nouveau plugin EMF com.mia.itmf.vbnet.metamodel v2.3.0

#### 1.1.11 Parser XML v2.0.4

- v2.0.4 (21/09/2020)
  - parsing SAX : lecture texte (evt characters) contreter découpage lié à caractères spéciaux ("<>& - &quot;&apos;&lt;&gt;&amp;) pour ne restituer qu'un unique XMLText
- v2.0.3 (10/07/2020)
  - console élimination double message sur fichier absent
  - ruleContext DumpXML : passage en mode bin + gestion écriture fichier nom long  
=> 260 cars (nécessite chemin absolu)