数据字典Oracle数据结构抓取工具v1.0

功能更新说明文档

## 功能描述:

1. 数据字典Oracle数据结构抓取工具v1.0（以下简称本工具）原有功能：
2. 用户可以配置导出系统的名称。
3. 用户可以配置导出系统的上线版本。
4. 用户可以配置导出文件的格式和路径。
5. 用户可以配置系统使用的数据库数量和每个数据库的相关配置，比如：IP地址、用户名、密码等。
6. 连接Oracle数据库抓取用户指定的库内全部的对象信息，包括表、视图、存储过程、Sequnece等类型，将抓取的对象信息按模板格式生成Excel文件。
7. 功能调整方案：
8. 原导出工具默认导出指定数据库的全部类型的对象定义信息，此次调整，增加导出对象类型的设置功能，默认导出设置为：表、视图，其他对象类型（序列INDEX、存储过程PROCEDURE、函数FUNCTION、触发器TRIGGER、索引INDEX、约束CONSTRAINT）可通过增加设置项的方式导出；

只导出表，视图，索引

1. 在支持按类型导出数据库对象的基础上，支持按导出对象列表导出的功能，同时，通过配置开关，该对象列表可标记为导出对象范围和非导出对象范围。
2. 在检出之前会判断配置信息中对象依赖的合法性。如要导出视图，则会判断视图依赖的表是否也在要导出的表配置中，如果不在则会提示用户增加对应的表配置。

## 二、配置说明：

用户需在本工具的dmet.properties文件里配置，以配置表为例：

1．在 dmet.properties 文件里的配置格式如下：

dmet.export.table .X=

dmet.export.tableInclude.X=

1. 配置项说明：
2. .X 是个可变的数值，根据用户配置了多少个数据库配置。
3. dmet.export.table.X ：

配置要导出或者非导出的表名称，用逗号隔开。

如果配置了空，则不会导出任何表。

如果配置 \* (表示全部)或者具体的表名 ，则根据dmet.export.tableInclude.X配置进行判断，默认值为 \* 。

1. dmet.export.tableInclude.X ：导出表还是非出表。

如果配置成Y ，则会导出dmet.export.table.X 中配置的表；

如果配置成N ，则会不导出dmet.export.table.X 中配置的表。

默认值是N 。

注意，如果dmet.export.table.X 配置了空 ，则此配置失效。

1. 其他如序列、存储过程、函数、触发器、索引、约束和表的配置类似。

## 配置举例：

下面的举例是在只导出一个库的前提下的。

1. 导出所有表、视图。配置如下：

dmet.export.table.1= \*

dmet.export.tableInclude.1 = Y

dmet.export.view.1= \*

dmet.export.viewInclude.1 = Y

2. 只导出所有表，配置如下：

dmet.export.table.1= \*

dmet.export.tableInclude.1 = Y

dmet.export.view.1 =

dmet.export.viewInclude.1 = N

3. 导出表名为t1,t2的表和视图名为v1, v2的视图，配置如下：

dmet.export.table.1= t1,t2

dmet.export.tableInclude.1 = Y

dmet.export.view.1 = v1,v2

dmet.export.viewInclude.1 = Y

4. 导出表名表名为t1,t2的表和导出除视图名为v1，v2的其他视图，配置如下：

dmet.export.table.1= t1,t2

dmet.export.tableInclude.1 = Y

dmet.export.view.1 = v1,v2

dmet.export.viewInclude.1 = N

5. 导出所有表和视图，还有存储过程名称为p1，p2的存储过程，配置如下：

dmet.export.table.1= \*

dmet.export.tableInclude.1 = Y

dmet.export.view.1= \*

dmet.export.viewInclude.1 = Y

dmet.export.procedure.1= p1,p2

dmet.export.procedure.1 = Y

6. 不导出表、视图和存储过程，只导出序列，配置如下：

dmet.export.table.1 =

dmet.export.tableInclude.1 = N

dmet.export.view.1 =

dmet.export.viewInclude.1 = Y

dmet.export.procedure.1=

dmet.export.procedureInclude.1 = Y

dmet.export.sequence.1= \*

dmet.export.sequenceInclude.1 = Y