

SQL Server DB (dbreport) Report : SQL Server Objects Report

**Report built on Dec 11 2005 11:24AM (local
time), Dec 11 2005 10:24AM (utc time)**

Server : SPAREE500

Service Name : MSSQLSERVER

Database Name : dbreport

**dbo DB user
Adrien Sales, Sourceforge**

SQL Server DB (dbreport) Report : SQL Server Objects Report :

Report built on Dec 11 2005 11:24AM (local time), Dec 11 2005 10:24AM (utc time)

Server : SPAREE500

Service Name : MSSQLSERVER

Database Name : dbreport

by dbo DB user and Adrien Sales

Table of Contents

1. SQLServer Reports Project Info	1
Details	1
2. GNU General Public License	2
Preamble	2
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION	2
Section 0	2
Section 1	3
Section 2	3
Section 3	4
Section 4	4
Section 5	4
Section 6	4
Section 7	4
Section 8	5
Section 9	5
Section 10	5
NO WARRANTY Section 11	5
Section 12	6
How to Apply These Terms to Your New Programs	6
3. Tables Report (3 tables)	8
dtproperties Table	8
Summary	8
Columns	8
Indexes and Primary Keys	8
titi Table	9
Summary	9
Columns	9
Products Table	9
Summary	9
Columns	10
Indexes and Primary Keys	10
4. Views (1 Views)	12
vw_toto	12
Summary	12
Columns	12
Source Code	12
5. Procedures (62 procs.)	14
dt_addtosourcecontrol	14
Summary	14
Parameters	14
Query Analyzer Call example	14
Source Code	14
dt_addtosourcecontrol_u	17
Summary	17
Parameters	17
Query Analyzer Call example	17
Source Code	18
dt_adduserobject	20
Summary	20
Parameters	21
Query Analyzer Call example	21
Source Code	21
dt_adduserobject_vcs	21

SQL Server DB (dbreport) Report :
SQL Server Objects Report

Summary	21
Parameters	22
Query Analyzer Call example	22
Source Code	22
dt_checkinobject	23
Summary	23
Parameters	23
Query Analyzer Call example	23
Source Code	23
dt_checkinobject_u	26
Summary	26
Parameters	27
Query Analyzer Call example	27
Source Code	27
dt_checkoutobject	30
Summary	30
Parameters	30
Query Analyzer Call example	31
Source Code	31
dt_checkoutobject_u	33
Summary	33
Parameters	33
Query Analyzer Call example	33
Source Code	33
dt_displayoerror	35
Summary	35
Parameters	35
Query Analyzer Call example	36
Source Code	36
dt_displayoerror_u	36
Summary	36
Parameters	37
Query Analyzer Call example	37
Source Code	37
dt_droppropertiesbyid	38
Summary	38
Parameters	38
Query Analyzer Call example	38
Source Code	38
dt_dropuserobjectbyid	39
Summary	39
Parameters	39
Query Analyzer Call example	39
Source Code	39
dt_generateansiname	40
Summary	40
Parameters	40
Query Analyzer Call example	40
Source Code	40
dt_getobjwithprop	41
Summary	41
Parameters	41
Query Analyzer Call example	42
Source Code	42
dt_getobjwithprop_u	43
Summary	43
Parameters	43
Query Analyzer Call example	43

SQL Server DB (dbreport) Report :
SQL Server Objects Report

Source Code	43
dt_getpropertiesbyid	44
Summary	44
Parameters	44
Query Analyzer Call example	44
Source Code	44
dt_getpropertiesbyid_u	45
Summary	45
Parameters	45
Query Analyzer Call example	45
Source Code	46
dt_getpropertiesbyid_vcs	46
Summary	46
Parameters	47
Query Analyzer Call example	47
Source Code	47
dt_getpropertiesbyid_vcs_u	48
Summary	48
Parameters	48
Query Analyzer Call example	48
Source Code	48
dt_isundersourcecontrol	49
Summary	49
Parameters	49
Query Analyzer Call example	49
Source Code	49
dt_isundersourcecontrol_u	51
Summary	51
Parameters	51
Query Analyzer Call example	51
Source Code	52
dt_removefromsourcecontrol	53
Summary	53
Parameters	54
Query Analyzer Call example	54
Source Code	54
dt_setpropertybyid	54
Summary	54
Parameters	55
Query Analyzer Call example	55
Source Code	55
dt_setpropertybyid_u	56
Summary	56
Parameters	56
Query Analyzer Call example	56
Source Code	56
dt_validateloginparams	58
Summary	58
Parameters	58
Query Analyzer Call example	58
Source Code	58
dt_validateloginparams_u	59
Summary	59
Parameters	59
Query Analyzer Call example	60
Source Code	60
dt_vcsenabled	61
Summary	61

SQL Server DB (dbreport) Report :
SQL Server Objects Report

Parameters	61
Query Analyzer Call example	61
Source Code	61
dt_verstamp006	62
Summary	62
Parameters	62
Query Analyzer Call example	62
Source Code	62
dt_whocheckedout	63
Summary	63
Parameters	63
Query Analyzer Call example	63
Source Code	63
dt_whocheckedout_u	65
Summary	65
Parameters	65
Query Analyzer Call example	65
Source Code	65
sp_dbk_print_book	67
Summary	67
Parameters	67
Query Analyzer Call example	67
Source Code	67
sp_dbk_print_bookinfo	69
Summary	69
Parameters	69
Query Analyzer Call example	69
Source Code	69
sp_dbk_print_extended_procedure_section	71
Summary	71
Parameters	71
Query Analyzer Call example	71
Source Code	71
sp_dbk_print_extended_procedures_chapter	73
Summary	73
Parameters	73
Query Analyzer Call example	73
Source Code	73
sp_dbk_print_filegroup_section	74
Summary	74
Parameters	74
Query Analyzer Call example	75
Source Code	75
sp_dbk_print_filegroups_book	77
Summary	77
Parameters	77
Query Analyzer Call example	77
Source Code	78
sp_dbk_print_filegroups_chapter	79
Summary	79
Parameters	79
Query Analyzer Call example	79
Source Code	79
sp_dbk_print_function_section	81
Summary	81
Parameters	81
Query Analyzer Call example	81
Source Code	81

SQL Server DB (dbreport) Report :
SQL Server Objects Report

sp_dbk_print_functions_book	84
Summary	84
Parameters	84
Query Analyzer Call example	85
Source Code	85
sp_dbk_print_functions_chapter	86
Summary	86
Parameters	86
Query Analyzer Call example	86
Source Code	86
sp_dbk_print_gpl_book	88
Summary	88
Parameters	88
Query Analyzer Call example	88
Source Code	88
sp_dbk_print_gpl_chapter	89
Summary	89
Parameters	89
Query Analyzer Call example	89
Source Code	89
sp_dbk_print_header	96
Summary	96
Parameters	96
Query Analyzer Call example	96
Source Code	96
sp_dbk_print_procedure_section	97
Summary	97
Parameters	97
Query Analyzer Call example	98
Source Code	98
sp_dbk_print_procedures_book	102
Summary	102
Parameters	102
Query Analyzer Call example	102
Source Code	102
sp_dbk_print_procedures_chapter	103
Summary	103
Parameters	103
Query Analyzer Call example	103
Source Code	103
sp_dbk_print_scripts_chapter	105
Summary	105
Parameters	105
Query Analyzer Call example	105
Source Code	105
sp_dbk_print_scripts_tables_stats	106
Summary	106
Parameters	106
Query Analyzer Call example	106
Source Code	106
sp_dbk_print_table_book	107
Summary	107
Parameters	108
Query Analyzer Call example	108
Source Code	108
sp_dbk_print_table_section	109
Summary	109
Parameters	109

SQL Server DB (dbreport) Report :
SQL Server Objects Report

Query Analyzer Call example	109
Source Code	110
sp_dbk_print_tables_book	115
Summary	116
Parameters	116
Query Analyzer Call example	116
Source Code	116
sp_dbk_print_tables_chapter	117
Summary	117
Parameters	117
Query Analyzer Call example	117
Source Code	117
sp_dbk_print_types_chapter	119
Summary	119
Parameters	119
Query Analyzer Call example	119
Source Code	119
sp_dbk_print_user_section	122
Summary	122
Parameters	122
Query Analyzer Call example	122
Source Code	122
sp_dbk_print_users_chapter	124
Summary	124
Parameters	125
Query Analyzer Call example	125
Source Code	125
sp_dbk_print_view_section	126
Summary	126
Parameters	126
Query Analyzer Call example	126
Source Code	126
sp_dbk_print_views_book	130
Summary	130
Parameters	130
Query Analyzer Call example	130
Source Code	130
sp_dbk_print_views_chapter	131
Summary	132
Parameters	132
Query Analyzer Call example	132
Source Code	132
sp_print_dbk_project_details_section	133
Summary	133
Parameters	133
Query Analyzer Call example	134
Source Code	134
sp_print_dbk_project_info_chapter	135
Summary	135
Parameters	135
Query Analyzer Call example	135
Source Code	135
sp_txt_print_gpl	136
Summary	136
Parameters	136
Query Analyzer Call example	136
Source Code	136
toto	142

SQL Server DB (dbreport) Report :
SQL Server Objects Report

Summary	142
Parameters	142
Query Analyzer Call example	142
Source Code	142
6. Functions (4 functs.)	144
test_function	144
Summary	144
Input Parameters (1)	144
Output Parameter	144
Source Code	144
fx_test	145
Summary	145
Input Parameters (1)	145
Output Parameter	145
Source Code	145
fx_test2	146
Summary	146
Input Parameters (2)	146
Output Parameter	146
Source Code	146
fx_dbk_get_xmlentities	147
Summary	147
Input Parameters (1)	147
Output Parameter	147
Source Code	147
7. Types (26)	149
List	149
8. Storage (1 filegroups)	151
PRIMARY filegroup	151
Summary	151
Data Files	151
9. Users(3)	152
AdrienS	152
dbo	152
toto	152
10. Management scripts	153
Tables Statistics Scripts	153

List of Tables

3.1. dtproperties Columns Details	8
3.2. Index pk_dtproperties Columns Details	9
3.3. titi Columns Details	9
3.4. Products Columns Details	10
3.5. Index ProductName Columns Details	11
3.6. Index toto Columns Details	11
4.1. vw_toto Columns Details	12
7.1. Types	149
9.1. AdrienS Details	152
9.2. dbo Details	152
9.3. toto Details	152

Chapter 1. SQLServer Reports Project Info

Details

SQLServer Reports [<http://sqlserverbook.sourceforge.net/>] is an open source project hosted by Sourceforge [<http://sourceforge.net/>], created and maintained by Adrien Sales.

Chapter 2. GNU General Public License

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software - to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps:

1. copyright the software, and
2. offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

Section 0

This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or

any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term “modification”.) Each licensee is addressed as “you”.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

Section 1

You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

Section 2

You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a. You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b. You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c. If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: If the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

Section 3

You may copy and distribute the Program (or a work based on it, under Section 2 in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a. Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b. Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c. Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

Section 4

You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

Section 5

You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

Section 6

Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

Section 7

If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

Section 8

If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

Section 9

The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

Section 10

If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY Section 11

BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTH-

ERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

Section 12

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.> Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type “show w”. This is free software, and you are welcome to redistribute it under certain conditions; type “show c” for details.

The hypothetical commands “show w” and “show c” should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than “show w” and “show c”; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a “copyright disclaimer” for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program “Gnomovision” (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989 Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

Chapter 3. Tables Report (3 tables)

dtproperties Table

Summary

Summarized details :

- Filegroup : PRIMARY
- Creation Date : Dec 8 2005 2:53PM
- (Max)Table Row Size (should not be greater than 8096): 857

Columns

Table 3.1. dtproperties Columns Details

Column Name	Type	Computed	Length	Prec	Scale	Nullable	Trim Trailing Blanks	Fixed Length Null	Collation
id	int	False	4	10	0	No	NA	NA	NA
objectid	int	False	4	10	0	Yes	NA	NA	NA
property	varchar	False	64	0	0	No	NA	NA	NA
value	varchar	False	255	0	0	Yes	NA	NA	NA
uvalue	nvarchar	False	510	0	0	Yes	NA	NA	NA
lvalue	image	False	16	0	0	Yes	NA	NA	NA
version	int	False	4	10	0	No	NA	NA	NA

Indexes and Primary Keys

pk_dtproperties

Summary

- Filegroup : PRIMARY
- Nb Columns : 2
- Row count : 0
- Max Len : 886
- Max irow : 94

- Lock Flags : 0
- Clustered :

Columns Details

Below are details on the columns of the pk_dtproperties index :

Table 3.2. Index pk_dtproperties Columns Details

Column Name	Type	Column Id	Key Id	Asc Sorted ?
id	int	1	1	TODO
property	varchar	3	2	TODO

titi Table

Summary

Summarized details :

- Filegroup : PRIMARY
- Creation Date : Dec 8 2005 2:54PM
- (Max)Table Row Size (should not be greater than 8096): 4

Columns

Table 3.3. titi Columns Details

Column Name	Type	Com-puted	Length	Prec	Scale	Nullable	Trim Trailing Blanks	Fixed Length Null	Collation
i	int	False	4	10	0	Yes	NA	NA	NA

Products Table

Summary

Summarized details :

- Filegroup : PRIMARY
- Creation Date : Nov 12 2005 1:58PM

- (Max)Table Row Size (should not be greater than 8096): 147

Columns

Table 3.4. Products Columns Details

Column Name	Type	Computed	Length	Prec	Scale	Nullable	Trim Trailing Blanks	Fixed Length Null	Collation
ProductID	int	False	4	10	0	No	NA	NA	NA
Product-Name	nvarchar	False	80	0	0	No	NA	NA	NA
SupplierID	int	False	4	10	0	Yes	NA	NA	NA
Category-ID	int	False	4	10	0	Yes	NA	NA	NA
QuantityPerUnit	nvarchar	False	40	0	0	Yes	NA	NA	NA
UnitPrice	money	False	8	19	4	Yes	NA	NA	NA
UnitsIn-Stock	smallint	False	2	5	0	Yes	NA	NA	NA
UnitsOn-Order	smallint	False	2	5	0	Yes	NA	NA	NA
Reorder-Level	smallint	False	2	5	0	Yes	NA	NA	NA
Discontinued	bit	False	1	1	0	No	NA	NA	NA

Indexes and Primary Keys

ProductName

Summary

- Filegroup : PRIMARY
- Nb Columns : 2
- Row count : 0
- Max Len : 108
- Max irow : 114
- Lock Flags : 0
- Clustered :

Columns Details

Below are details on the columns of the ProductName index :

Table 3.5. Index ProductName Columns Details

Column Name	Type	Column Id	Key Id	Asc Sorted ?
ProductName	nvarchar	2	1	TODO

toto

Summary

- Filegroup : PRIMARY
- Nb Columns : 3
- Row count : 0
- Max Len : 112
- Max irow : 118
- Lock Flags : 0
- Clustered :

Columns Details

Below are details on the columns of the toto index :

Table 3.6. Index toto Columns Details

Column Name	Type	Column Id	Key Id	Asc Sorted ?
ProductID	int	1	1	TODO
ProductName	nvarchar	2	2	TODO

Chapter 4. Views (1 Views)

vw_toto

Summary

Find below summarized details of vw_toto :

- Creation Date : Dec 8 2005 2:07PM

:

Columns

Table 4.1. vw_toto Columns Details

Column Name	Type	Computed	Length	Prec	Nullable	Collation
ProductID	int	No	4	10	No	
ProductName	nvarchar	No	80	0	No	SQL_Latin1_General_CP1_CI_AS
SupplierID	int	No	4	10	Yes	
CategoryID	int	No	4	10	Yes	
QuantityPer-Unit	nvarchar	No	40	0	Yes	SQL_Latin1_General_CP1_CI_AS
UnitPrice	money	No	8	19	Yes	
UnitsInStock	smallint	No	2	5	Yes	
UnitsOnOrder	smallint	No	2	5	Yes	
ReorderLevel	smallint	No	2	5	Yes	
Discontinued	bit	No	1	1	No	

Source Code

Below find the TSQL statement used to create (or recreate)vw_toto view :

```
if exists (select * from dbo.sysobjects
where id = object_id(N'[dbo].[vw_toto]')
and OBJECTPROPERTY(id, N'IsView') = 1)
drop view [dbo].[vw_toto]
GO

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO
```

```
create view vw_toto as select * from Products  
GO  
SET QUOTED_IDENTIFIER OFF  
GO  
SET ANSI_NULLS ON  
GO
```

Chapter 5. Procedures (62 procs.)

dt_addtosourcecontrol

Summary

Find below summarized details :

- Creation Date : Dec 8 2005 2:53PM

Parameters

Find below input parameters :

1. @vchSourceSafeINI : varchar (255)
2. @vchProjectName : varchar (255)
3. @vchComment : varchar (255)
4. @vchLoginName : varchar (255)
5. @vchPassword : varchar (255)

Query Analyzer Call example

Find below how to call dt_addtosourcecontrol procedure :

/ Warning : do not forget to include declare section before to use the call example ! */*

```
exec dt_addtosourcecontrol @vchSourceSafeINI, @vchProjectName, @vchComment,  
@vchLoginName, @vchPassword
```

Source Code

Below find the TSQL statement used to create (or recreate)dt_addtosourcecontrol procedure :

```
if exists(  
select * from dbo.sysobjects  
where id = object_id(N'[dbo].[dt_addtosourcecontrol]')  
and OBJECTPROPERTY(id, N'IsProcedure') = 1)  
drop procedure [dbo].[dt_addtosourcecontrol]  
  
SET QUOTED_IDENTIFIER ON  
GO  
SET ANSI_NULLS ON  
GO
```

```
create proc dbo.dt_addtosourcecontrol
    @vchSourceSafeINI varchar(255) = '',
    @vchProjectName   varchar(255) = '',
    @vchComment       varchar(255) = '',
    @vchLoginName     varchar(255) = '',
    @vchPassword      varchar(255) = ''

as

set nocount on

declare @iReturn int
declare @iObjectId int
select @iObjectId = 0

declare @iStreamObjectId int
select @iStreamObjectId = 0

declare @VSSGUID varchar(100)
select @VSSGUID = 'SQLVersionControl.VCS_SQL'

declare @vchDatabaseName varchar(255)
select @vchDatabaseName = db_name()

declare @iReturnValue int
select @iReturnValue = 0

declare @iPropertyObjectId int
declare @vchParentId varchar(255)

declare @iObjectCount int
select @iObjectCount = 0

    exec @iReturn = sp_OACreate @VSSGUID, @iObjectId OUT
    if @iReturn <> 0 GOTO E_OAError

    /* Create Project in SS */
    exec @iReturn = sp_OAMethod @iObjectId,
        'AddProjectToSourceSafe',
        NULL,
        @vchSourceSafeINI,
        @vchProjectName output,
        @@SERVERNAME,
        @vchDatabaseName,
        @vchLoginName,
        @vchPassword,
        @vchComment

    if @iReturn <> 0 GOTO E_OAError

    exec @iReturn = sp_OAGetProperty @iObjectId, 'GetStreamObject', @iStreamObject

    if @iReturn <> 0 GOTO E_OAError

    /* Set Database Properties */
    begin tran SetProperties

    /* add high level object */

    exec @iPropertyObjectId = dbo.dt_adduserobject_vcs 'VCSProjectID'
```

```

select @vchParentId = CONVERT(varchar(255),@iPropertyObjectId)

exec dbo.dt_setpropertybyid @iPropertyObjectId, 'VCSProjectID', @vchParentId ,
exec dbo.dt_setpropertybyid @iPropertyObjectId, 'VCSProject' , @vchProjectName
exec dbo.dt_setpropertybyid @iPropertyObjectId, 'VCSSourceSafeINI' , @vchSourceSafeINI
exec dbo.dt_setpropertybyid @iPropertyObjectId, 'VCSSQLServer', @@SERVERNAME,
exec dbo.dt_setpropertybyid @iPropertyObjectId, 'VCSSQLDatabase', @vchDatabaseName

if @@error <> 0 GOTO E_General_Error

commit tran SetProperties

declare cursorProcNames cursor for
    select convert(varchar(255), name) from sysobjects where type = 'P' and name <> ''
open cursorProcNames

while 1 = 1
begin
    declare @vchProcName varchar(255)
    fetch next from cursorProcNames into @vchProcName
    if @@fetch_status <> 0
        break

    select colid, text into #ProcLines
    from syscomments
    where id = object_id(@vchProcName)
    order by colid

    declare @iCurProcLine int
    declare @iProcLines int
    select @iCurProcLine = 1
    select @iProcLines = (select count(*) from #ProcLines)
    while @iCurProcLine <= @iProcLines
    begin
        declare @pos int
        select @pos = 1
        declare @iCurLineSize int
        select @iCurLineSize = len((select text from #ProcLines where colid = @iCurProcLine))
        while @pos <= @iCurLineSize
        begin
            declare @vchProcLinePiece varchar(255)
            select @vchProcLinePiece = convert(varchar(255),
                substring((select text from #ProcLines where colid = @iCurProcLine)
                    , @pos, 255 ))
            exec @iReturn = sp_OAMethod @iStreamObjectId, 'AddStream', @iReturn, @vchProcLinePiece
            if @iReturn <> 0 GOTO E_OAError
            select @pos = @pos + 255
        end
        select @iCurProcLine = @iCurProcLine + 1
    end
    drop table #ProcLines

    exec @iReturn = sp_OAMethod @iObjectId,
        'CheckIn_StoredProcedure',
        NULL,
        @sProjectName = @vchProjectName,
        @sSourceSafeINI = @vchSourceSafeINI,
        @sServerName = @@SERVERNAME,
        @sDatabaseName = @vchDatabaseName,
        @sObjectName = @vchProcName,
        @sComment = @vchComment,
        @sLoginName = @vchLoginName,
        @sPassword = @vchPassword,
        @iVCSFlags = 0,

```

```

                                @iActionFlag = 0,
                                @sStream = ''
                                if @iReturn = 0 select @iObjectCount = @iObjectCount + 1
                                end
CleanUp:
    close cursorProcNames
    deallocate cursorProcNames
    select @vchProjectName
    select @iObjectCount
    return
E_General_Error:
    /* this is an all or nothing.  No specific error messages */
    goto CleanUp
E_OAError:
    exec dbo.dt_displayoerror @iObjectId, @iReturn
    goto CleanUp

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

dt_addtosourcecontrol_u

Summary

Find below summarized details :

- Creation Date : Dec 8 2005 2:53PM

Parameters

Find below input parameters :

1. @vchSourceSafeINI : nvarchar (510)
2. @vchProjectName : nvarchar (510)
3. @vchComment : nvarchar (510)
4. @vchLoginName : nvarchar (510)
5. @vchPassword : nvarchar (510)

Query Analyzer Call example

Find below how to call dt_addtosourcecontrol_u procedure :

```
/* Warning : do not forget to include declare section before to use the call example ! */
```

```
exec dt_addtosourcecontrol_u @vchSourceSafeINI, @vchProjectName, @vchComment,
@vchLoginName, @vchPassword
```

Source Code

Below find the TSQL statement used to create (or recreate)dt_addtosourcecontrol_u procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[dt_addtosourcecontrol_u]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[dt_addtosourcecontrol_u]
```

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO
```

```
create proc dbo.dt_addtosourcecontrol_u
    @vchSourceSafeINI nvarchar(255) = '',
    @vchProjectName nvarchar(255) = '',
    @vchComment nvarchar(255) = '',
    @vchLoginName nvarchar(255) = '',
    @vchPassword nvarchar(255) = ''
```

```
as
```

```
set nocount on
```

```
declare @iReturn int
declare @iObjectId int
select @iObjectId = 0
```

```
declare @iStreamObjectId int
select @iStreamObjectId = 0
```

```
declare @VSSGUID nvarchar(100)
select @VSSGUID = N'SQLVersionControl.VCS_SQL'
```

```
declare @vchDatabaseName varchar(255)
select @vchDatabaseName = db_name()
```

```
declare @iReturnValue int
select @iReturnValue = 0
```

```
declare @iPropertyObjectId int
declare @vchParentId nvarchar(255)
```

```
declare @iObjectCount int
select @iObjectCount = 0
```

```
exec @iReturn = sp_OACreate @VSSGUID, @iObjectId OUT
if @iReturn <> 0 GOTO E_OAError
```

```
/* Create Project in SS */
exec @iReturn = sp_OAMethod @iObjectId,
```

```

        'AddProjectToSourceSafe',
        NULL,
        @vchSourceSafeINI,
        @vchProjectName output,
        @@SERVERNAME,
        @vchDatabaseName,
        @vchLoginName,
        @vchPassword,
        @vchComment

if @iReturn <> 0 GOTO E_OAError

exec @iReturn = sp_OAGetProperty @iObjectId, N'GetStreamObject', @iStreamObjec

if @iReturn <> 0 GOTO E_OAError

/* Set Database Properties */

begin tran SetProperties

/* add high level object */

exec @iPropertyObjectId = dbo.dt_adduserobject_vcs 'VCSPROJECTID'

select @vchParentId = CONVERT(nvarchar(255),@iPropertyObjectId)

exec dbo.dt_setpropertybyid_u @iPropertyObjectId, 'VCSPROJECTID', @vchParentId
exec dbo.dt_setpropertybyid_u @iPropertyObjectId, 'VCSPROJECT', @vchProjectName
exec dbo.dt_setpropertybyid_u @iPropertyObjectId, 'VCSSOURCESAFEINI', @vchSourceSafeINI
exec dbo.dt_setpropertybyid_u @iPropertyObjectId, 'VCSSQLSERVER', @@SERVERNAME
exec dbo.dt_setpropertybyid_u @iPropertyObjectId, 'VCSSQLDATABASE', @vchDatabaseName

if @@error <> 0 GOTO E_General_Error

commit tran SetProperties

declare cursorProcNames cursor for
    select convert(nvarchar(255), name) from sysobjects where type = N'P' and
open cursorProcNames

while 1 = 1
begin
    declare @vchProcName nvarchar(255)
    fetch next from cursorProcNames into @vchProcName
    if @@fetch_status <> 0
        break

    select colid, text into #ProcLines
    from syscomments
    where id = object_id(@vchProcName)
    order by colid

    declare @iCurProcLine int
    declare @iProcLines int
    select @iCurProcLine = 1
    select @iProcLines = (select count(*) from #ProcLines)
    while @iCurProcLine <= @iProcLines
    begin
        declare @pos int
        select @pos = 1
        declare @iCurLineSize int
        select @iCurLineSize = len((select text from #ProcLines where colid =
        while @pos <= @iCurLineSize

```

```

begin
    declare @vchProcLinePiece nvarchar(255)
    select @vchProcLinePiece = convert(nvarchar(255),
        substring((select text from #ProcLines where colid = @iCurProc
            @pos, 255 ))
    exec @iReturn = sp_OAMethod @iStreamObjectId, N'AddStream', @iRetu
    if @iReturn <> 0 GOTO E_OAError
    select @pos = @pos + 255
end
select @iCurProcLine = @iCurProcLine + 1
end
drop table #ProcLines

exec @iReturn = sp_OAMethod @iObjectId,
    'CheckIn_StoredProcedure',
    NULL,
    @sProjectName = @vchProjectName,
    @sSourceSafeINI = @vchSourceSafeINI,
    @sServerName = @@SERVERNAME,
    @sDatabaseName = @vchDatabaseName,
    @sObjectName = @vchProcName,
    @sComment = @vchComment,
    @sLoginName = @vchLoginName,
    @sPassword = @vchPassword,
    @iVCSFlags = 0,
    @iActionFlag = 0,
    @sStream = ''

    if @iReturn = 0 select @iObjectCount = @iObjectCount + 1
end

CleanUp:
    close cursorProcNames
    deallocate cursorProcNames
    select @vchProjectName
    select @iObjectCount
    return

E_General_Error:
    /* this is an all or nothing.  No specific error messages */
    goto CleanUp

E_OAError:
    exec dbo.dt_displayoerror_u @iObjectId, @iReturn
    goto CleanUp

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

```

dt_adduserobject

Summary

Find below summarized details :

- Creation Date : Dec 8 2005 2:53PM

Parameters

No Input Parameters Required

Query Analyzer Call example

Find below how to call dt_adduserobject procedure :

```
exec dt_adduserobject
```

Source Code

Below find the TSQL statement used to create (or recreate)dt_adduserobject procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[dt_adduserobject]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[dt_adduserobject]

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

/*
** Add an object to the dtproperties table
*/
create procedure dbo.dt_adduserobject
as
    set nocount on
    /*
    ** Create the user object if it does not exist already
    */
    begin transaction
        insert dbo.dtproperties (property) VALUES ('DtgSchemaOBJECT')
        update dbo.dtproperties set objectid=@@identity
            where id=@@identity and property='DtgSchemaOBJECT'
    commit
    return @@identity
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

dt_adduserobject_vcs

Summary

Find below summarized details :

- Creation Date : Dec 8 2005 2:53PM

Parameters

Find below input parameters :

1. @vchProperty : varchar (64)

Query Analyzer Call example

Find below how to call dt_adduserobject_vcs procedure :

/* Warning : do not forget to include declare section before to use the call example ! */

```
exec dt_adduserobject_vcs @vchProperty
```

Source Code

Below find the TSQL statement used to create (or recreate)dt_adduserobject_vcs procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[dt_adduserobject_vcs]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[dt_adduserobject_vcs]

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

create procedure dbo.dt_adduserobject_vcs
    @vchProperty varchar(64)

as

set nocount on

declare @iReturn int
/*
** Create the user object if it does not exist already
*/
begin transaction
    select @iReturn = objectid from dbo.dtproperties where property = @vchProp
    if @iReturn IS NULL
    begin
        insert dbo.dtproperties (property) VALUES (@vchProperty)
        update dbo.dtproperties set objectid=@@identity
        where id=@@identity and property=@vchProperty
        select @iReturn = @@identity
    end
commit
return @iReturn
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

dt_checkinobject

Summary

Find below summarized details :

- Creation Date : Dec 8 2005 2:53PM

Parameters

Find below input parameters :

1. @chObjectType : char (4)
2. @vchObjectName : varchar (255)
3. @vchComment : varchar (255)
4. @vchLoginName : varchar (255)
5. @vchPassword : varchar (255)
6. @iVCSFlags : int (4)
7. @iActionFlag : int (4)
8. @txStream1 : text (16)
9. @txStream2 : text (16)
10. @txStream3 : text (16)

Query Analyzer Call example

Find below how to call dt_checkinobject procedure :

```
/* Warning : do not forget to include declare section before to use the call example ! */
```

```
exec dt_checkinobject @chObjectType, @vchObjectName, @vchComment, @vchLoginName,
@vchPassword, @iVCSFlags, @iActionFlag, @txStream1, @txStream2, @txStream3
```

Source Code

Below find the TSQL statement used to create (or recreate)dt_checkinobject procedure :

```

if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[dt_checkinobject]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[dt_checkinobject]

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

create proc dbo.dt_checkinobject
    @chObjectType char(4),
    @vchObjectName varchar(255),
    @vchComment varchar(255)='',
    @vchLoginName varchar(255),
    @vchPassword varchar(255)='',
    @iVCSFlags int = 0,
    @iActionFlag int = 0, /* 0 => AddFile, 1 => CheckIn */
    @txStream1 Text = '', /* There is a bug that if items are NULL they do not
    @txStream2 Text = '',
    @txStream3 Text = ''

as

set nocount on

declare @iReturn int
declare @iObjectId int
select @iObjectId = 0

declare @VSSGUID varchar(100)
select @VSSGUID = 'SQLVersionControl.VCS_SQL'

declare @iPropertyObjectId int
select @iPropertyObjectId = 0

    select @iPropertyObjectId = (select objectid from dbo.dtproperties where prope

    declare @vchProjectName varchar(255)
    declare @vchSourceSafeINI varchar(255)
    declare @vchServerName varchar(255)
    declare @vchDatabaseName varchar(255)
    exec dbo.dt_getpropertiesbyid_vcs @iPropertyObjectId, 'VCSPROJECT', @vch
    exec dbo.dt_getpropertiesbyid_vcs @iPropertyObjectId, 'VCSSOURCESAFEINI', @vch
    exec dbo.dt_getpropertiesbyid_vcs @iPropertyObjectId, 'VCSSQLSERVER', @vch
    exec dbo.dt_getpropertiesbyid_vcs @iPropertyObjectId, 'VCSSQLDATABASE', @vch

    if @chObjectType = 'PROC'
    begin
        if @iActionFlag = 1
        begin
            /* Procedure Can have up to three streams
            Drop Stream, Create Stream, GRANT stream */

            begin tran compile_all

            /* try to compile the streams */
            exec (@txStream1)

```

```

        if @@error <> 0 GOTO E_Compile_Fail

        exec (@txStream2)
        if @@error <> 0 GOTO E_Compile_Fail

        exec (@txStream3)
        if @@error <> 0 GOTO E_Compile_Fail
    end

    exec @iReturn = sp_OACreate @VSSGUID, @iObjectId OUT
    if @iReturn <> 0 GOTO E_OAError

    if @iActionFlag = 1
    begin
        exec @iReturn = sp_OAMethod @iObjectId,
            'CheckIn_StoredProcedure',
            NULL,
            @sProjectName = @vchProjectName,
            @sSourceSafeINI = @vchSourceSafeINI,
            @sServerName = @vchServerName,
            @sDatabaseName = @vchDatabaseName,
            @sObjectName = @vchObjectName,
            @sComment = @vchComment,
            @sLoginName = @vchLoginName,
            @sPassword = @vchPassword,
            @iVCSFlags = @iVCSFlags,
            @iActionFlag = @iActionFlag,
            @sStream = @txStream2
    end
    else
    begin
        declare @iStreamObjectId int
        declare @iReturnValue int

        exec @iReturn = sp_OAGetProperty @iObjectId, 'GetStreamObject', @iStreamObjectId
        if @iReturn <> 0 GOTO E_OAError

        select colid, text into #ProcLines
        from syscomments
        where id = object_id(@vchObjectName)
        order by colid

        declare @iCurProcLine int
        declare @iProcLines int
        select @iCurProcLine = 1
        select @iProcLines = (select count(*) from #ProcLines)
        while @iCurProcLine <= @iProcLines
        begin
            declare @pos int
            select @pos = 1
            declare @iCurLineSize int
            select @iCurLineSize = len((select text from #ProcLines where colid = @iCurProcLine))
            while @pos <= @iCurLineSize
            begin
                declare @vchProcLinePiece varchar(255)
                select @vchProcLinePiece = convert(varchar(255),
                    substring((select text from #ProcLines where colid = @iCurProcLine),
                        @pos, 255 ))
                exec @iReturn = sp_OAMethod @iStreamObjectId, 'AddStream', @iStreamObjectId, @vchProcLinePiece
                if @iReturn <> 0 GOTO E_OAError
                select @pos = @pos + 255
            end
            select @iCurProcLine = @iCurProcLine + 1
        end
    end
end

```

```

drop table #ProcLines

exec @iReturn = sp_OAMethod @iObjectId,
    'CheckIn_StoredProcedure',
    NULL,
    @sProjectName = @vchProjectName,
    @sSourceSafeINI = @vchSourceSafeINI,
    @sServerName = @vchServerName,
    @sDatabaseName = @vchDatabaseName,
    @sObjectName = @vchObjectName,
    @sComment = @vchComment,
    @sLoginName = @vchLoginName,
    @sPassword = @vchPassword,
    @iVCSFlags = @iVCSFlags,
    @iActionFlag = @iActionFlag,
    @sStream = ''

end

if @iReturn <> 0 GOTO E_OAError

if @iActionFlag = 1
begin
    commit tran compile_all
    if @@error <> 0 GOTO E_Compile_Fail
end

end

CleanUp:
    return

E_Compile_Fail:
    declare @lerror int
    select @lerror = @@error
    rollback tran compile_all
    RAISERROR (@lerror,16,-1)
    goto CleanUp

E_OAError:
    if @iActionFlag = 1 rollback tran compile_all
    exec dbo.dt_displayoaerror @iObjectId, @iReturn
    goto CleanUp

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

```

dt_checkinobject_u

Summary

Find below summarized details :

- Creation Date : Dec 8 2005 2:53PM

Parameters

Find below input parameters :

1. @chObjectType : char (4)
2. @vchObjectName : nvarchar (510)
3. @vchComment : nvarchar (510)
4. @vchLoginName : nvarchar (510)
5. @vchPassword : nvarchar (510)
6. @iVCSFlags : int (4)
7. @iActionFlag : int (4)
8. @txStream1 : text (16)
9. @txStream2 : text (16)
10. @txStream3 : text (16)

Query Analyzer Call example

Find below how to call dt_checkinobject_u procedure :

```
/* Warning : do not forget to include declare section before to use the call example ! */
```

```
exec dt_checkinobject_u @chObjectType, @vchObjectName, @vchComment, @vchLoginName,  
@vchPassword, @iVCSFlags, @iActionFlag, @txStream1, @txStream2, @txStream3
```

Source Code

Below find the TSQL statement used to create (or recreate)dt_checkinobject_u procedure :

```
if exists(  
select * from dbo.sysobjects  
where id = object_id(N'[dbo].[dt_checkinobject_u]')  
and OBJECTPROPERTY(id, N'IsProcedure') = 1)  
drop procedure [dbo].[dt_checkinobject_u]
```

```
SET QUOTED_IDENTIFIER ON  
GO  
SET ANSI_NULLS ON  
GO
```

```
create proc dbo.dt_checkinobject_u  
    @chObjectType char(4),  
    @vchObjectName nvarchar(255),  
    @vchComment nvarchar(255)='',  
    @vchLoginName nvarchar(255),  
    @vchPassword nvarchar(255)='',  
    @iVCSFlags int = 0,
```

```

    @iActionFlag    int = 0,    /* 0 => AddFile, 1 => CheckIn */
    @txStream1      Text = '',  /* There is a bug that if items are NULL they do not
    @txStream2      Text = '',
    @txStream3      Text = ''

as

set nocount on

declare @iReturn int
declare @iObjectId int
select @iObjectId = 0

declare @VSSGUID nvarchar(100)
select @VSSGUID = N'SQLVersionControl.VCS_SQL'

declare @iPropertyObjectId int
select @iPropertyObjectId = 0

    select @iPropertyObjectId = (select objectid from dbo.dtproperties where prope

declare @vchProjectName    nvarchar(255)
declare @vchSourceSafeINI nvarchar(255)
declare @vchServerName    nvarchar(255)
declare @vchDatabaseName  nvarchar(255)
exec dbo.dt_getpropertiesbyid_vcs_u @iPropertyObjectId, 'VCSProject',    @v
exec dbo.dt_getpropertiesbyid_vcs_u @iPropertyObjectId, 'VCSSourceSafeINI', @v
exec dbo.dt_getpropertiesbyid_vcs_u @iPropertyObjectId, 'VCSSQLServer',    @v
exec dbo.dt_getpropertiesbyid_vcs_u @iPropertyObjectId, 'VCSSQLDatabase',  @v

if @chObjectType = 'PROC'
begin
    if @iActionFlag = 1
    begin
        /* Procedure Can have up to three streams
        Drop Stream, Create Stream, GRANT stream */

        begin tran compile_all

            /* try to compile the streams */
            exec (@txStream1)
            if @@error <> 0 GOTO E_Compile_Fail

            exec (@txStream2)
            if @@error <> 0 GOTO E_Compile_Fail

            exec (@txStream3)
            if @@error <> 0 GOTO E_Compile_Fail
        end

        exec @iReturn = sp_OACreate @VSSGUID, @iObjectId OUT
        if @iReturn <> 0 GOTO E_OAError

        if @iActionFlag = 1
        begin
            exec @iReturn = sp_OAMethod @iObjectId,
                N'CheckIn_StoredProcedure',
                NULL,
                @sProjectName = @vchProjectName,
                @sSourceSafeINI = @vchSourceSafeINI,
                @sServerName = @vchServerName,
                @sDatabaseName = @vchDatabaseName,

```

```

                                @sObjectName = @vchObjectName,
                                @sComment = @vchComment,
                                @sLoginName = @vchLoginName,
                                @sPassword = @vchPassword,
                                @iVCSFlags = @iVCSFlags,
                                @iActionFlag = @iActionFlag,
                                @sStream = @txStream2
end
else
begin
    declare @iStreamObjectId int
    declare @iReturnValue int

    exec @iReturn = sp_OAGetProperty @iObjectId, N'GetStreamObject', @iStream
    if @iReturn <> 0 GOTO E_OAError

    select colid, text into #ProcLines
    from syscomments
    where id = object_id(@vchObjectName)
    order by colid

    declare @iCurProcLine int
    declare @iProcLines int
    select @iCurProcLine = 1
    select @iProcLines = (select count(*) from #ProcLines)
    while @iCurProcLine <= @iProcLines
    begin
        declare @pos int
        select @pos = 1
        declare @iCurLineSize int
        select @iCurLineSize = len((select text from #ProcLines where colid = @iCurProcLine))
        while @pos <= @iCurLineSize
        begin
            declare @vchProcLinePiece nvarchar(255)
            select @vchProcLinePiece = convert(nvarchar(255),
                substring((select text from #ProcLines where colid = @iCurProcLine),
                    @pos, 255 ))
            exec @iReturn = sp_OAMethod @iStreamObjectId, N'AddStream', @iStream, @vchProcLinePiece
            if @iReturn <> 0 GOTO E_OAError
            select @pos = @pos + 255
        end
        select @iCurProcLine = @iCurProcLine + 1
    end
    drop table #ProcLines

    exec @iReturn = sp_OAMethod @iObjectId,
                                N'CheckIn_StoredProcedure',
                                NULL,
                                @sProjectName = @vchProjectName,
                                @sSourceSafeINI = @vchSourceSafeINI,
                                @sServerName = @vchServerName,
                                @sDatabaseName = @vchDatabaseName,
                                @sObjectName = @vchObjectName,
                                @sComment = @vchComment,
                                @sLoginName = @vchLoginName,
                                @sPassword = @vchPassword,
                                @iVCSFlags = @iVCSFlags,
                                @iActionFlag = @iActionFlag,
                                @sStream = ''
end

if @iReturn <> 0 GOTO E_OAError

if @iActionFlag = 1

```



```
        begin
            commit tran compile_all
            if @@error <> 0 GOTO E_Compile_Fail
        end

    end

CleanUp:
    return

E_Compile_Fail:
    declare @lerror int
    select @lerror = @@error
    rollback tran compile_all
    RAISERROR (@lerror,16,-1)
    goto CleanUp

E_OAError:
    if @iActionFlag = 1 rollback tran compile_all
    exec dbo.dt_displayoaerror_u @iObjectId, @iReturn
    goto CleanUp

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

dt_checkoutobject

Summary

Find below summarized details :

- Creation Date : Dec 8 2005 2:53PM

Parameters

Find below input parameters :

1. @chObjectType : char (4)
2. @vchObjectName : varchar (255)
3. @vchComment : varchar (255)
4. @vchLoginName : varchar (255)
5. @vchPassword : varchar (255)
6. @iVCSFlags : int (4)
7. @iActionFlag : int (4)

Query Analyzer Call example

Find below how to call dt_checkoutobject procedure :

```
/* Warning : do not forget to include declare section before to use the call example ! */
```

```
exec dt_checkoutobject @chObjectType, @vchObjectName, @vchComment, @vchLoginName,
@vchPassword, @iVCSFlags, @iActionFlag
```

Source Code

Below find the TSQL statement used to create (or recreate)dt_checkoutobject procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[dt_checkoutobject]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[dt_checkoutobject]

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

create proc dbo.dt_checkoutobject
    @chObjectType char(4),
    @vchObjectName varchar(255),
    @vchComment varchar(255),
    @vchLoginName varchar(255),
    @vchPassword varchar(255),
    @iVCSFlags int = 0,
    @iActionFlag int = 0/* 0 => Checkout, 1 => GetLatest, 2 => UndoCheckOut */

as

set nocount on

declare @iReturn int
declare @iObjectId int
select @iObjectId =0

declare @VSSGUID varchar(100)
select @VSSGUID = 'SQLVersionControl.VCS_SQL'

declare @iReturnValue int
select @iReturnValue = 0

declare @vchTempText varchar(255)

/* this is for our strings */
declare @iStreamObjectId int
select @iStreamObjectId = 0

    declare @iPropertyObjectId int
    select @iPropertyObjectId = (select objectid from dbo.dtproperties where prop

    declare @vchProjectName varchar(255)
    declare @vchSourceSafeINI varchar(255)
```

```

declare @vchServerName    varchar(255)
declare @vchDatabaseName  varchar(255)
exec dbo.dt_getpropertiesbyid_vcs @iPropertyObjectId, 'VCSProject', @vch
exec dbo.dt_getpropertiesbyid_vcs @iPropertyObjectId, 'VCSSourceSafeINI', @vch
exec dbo.dt_getpropertiesbyid_vcs @iPropertyObjectId, 'VCSSQLServer', @vch
exec dbo.dt_getpropertiesbyid_vcs @iPropertyObjectId, 'VCSSQLDatabase', @vch

if @chObjectType = 'PROC'
begin
    /* Procedure Can have up to three streams
       Drop Stream, Create Stream, GRANT stream */

    exec @iReturn = sp_OACreate @VSSGUID, @iObjectId OUT

    if @iReturn <> 0 GOTO E_OAError

    exec @iReturn = sp_OAMethod @iObjectId,
        'CheckOut_StoredProcedure',
        NULL,
        @sProjectName = @vchProjectName,
        @sSourceSafeINI = @vchSourceSafeINI,
        @sObjectName = @vchObjectName,
        @sServerName = @vchServerName,
        @sDatabaseName = @vchDatabaseName,
        @sComment = @vchComment,
        @sLoginName = @vchLoginName,
        @sPassword = @vchPassword,
        @iVCSFlags = @iVCSFlags,
        @iActionFlag = @iActionFlag

    if @iReturn <> 0 GOTO E_OAError

    exec @iReturn = sp_OAGetProperty @iObjectId, 'GetStreamObject', @iStreamOb

    if @iReturn <> 0 GOTO E_OAError

    create table #commenttext (id int identity, sourcecode varchar(255))

    select @vchTempText = 'STUB'
    while @vchTempText IS NOT NULL
    begin
        exec @iReturn = sp_OAMethod @iStreamObjectId, 'GetStream', @iReturnVal
        if @iReturn <> 0 GOTO E_OAError

        if (@vchTempText IS NOT NULL) insert into #commenttext (sourcecode) se
    end

    select 'VCS'=sourcecode from #commenttext order by id
    select 'SQL'=text from syscomments where id = object_id(@vchObjectName) or

end

CleanUp:
    return

E_OAError:
    exec dbo.dt_displayoaerror @iObjectId, @iReturn
    GOTO CleanUp

GO
SET QUOTED_IDENTIFIER OFF

```

```
GO
SET ANSI_NULLS ON
GO
```

dt_checkoutobject_u

Summary

Find below summarized details :

- Creation Date : Dec 8 2005 2:53PM

Parameters

Find below input parameters :

1. @chObjectType : char (4)
2. @vchObjectName : nvarchar (510)
3. @vchComment : nvarchar (510)
4. @vchLoginName : nvarchar (510)
5. @vchPassword : nvarchar (510)
6. @iVCSFlags : int (4)
7. @iActionFlag : int (4)

Query Analyzer Call example

Find below how to call dt_checkoutobject_u procedure :

```
/* Warning : do not forget to include declare section before to use the call example ! */
```

```
exec dt_checkoutobject_u @chObjectType, @vchObjectName, @vchComment, @vchLoginName,
@vchPassword, @iVCSFlags, @iActionFlag
```

Source Code

Below find the TSQL statement used to create (or recreate)dt_checkoutobject_u procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[dt_checkoutobject_u]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[dt_checkoutobject_u]
```

```
SET QUOTED_IDENTIFIER ON
```

```

GO
SET ANSI_NULLS ON
GO

create proc dbo.dt_checkoutobject_u
    @chObjectType char(4),
    @vchObjectName nvarchar(255),
    @vchComment nvarchar(255),
    @vchLoginName nvarchar(255),
    @vchPassword nvarchar(255),
    @iVCSFlags int = 0,
    @iActionFlag int = 0/* 0 => Checkout, 1 => GetLatest, 2 => UndoCheckOut */
as

set nocount on

declare @iReturn int
declare @iObjectId int
select @iObjectId =0

declare @VSSGUID nvarchar(100)
select @VSSGUID = N'SQLVersionControl.VCS_SQL'

declare @iReturnValue int
select @iReturnValue = 0

declare @vchTempText nvarchar(255)

/* this is for our strings */
declare @iStreamObjectId int
select @iStreamObjectId = 0

    declare @iPropertyObjectId int
    select @iPropertyObjectId = (select objectid from dbo.dtproperties where prope

    declare @vchProjectName nvarchar(255)
    declare @vchSourceSafeINI nvarchar(255)
    declare @vchServerName nvarchar(255)
    declare @vchDatabaseName nvarchar(255)
    exec dbo.dt_getpropertiesbyid_vcs_u @iPropertyObjectId, 'VCSProject', @v
    exec dbo.dt_getpropertiesbyid_vcs_u @iPropertyObjectId, 'VCSSourceSafeINI', @v
    exec dbo.dt_getpropertiesbyid_vcs_u @iPropertyObjectId, 'VCSSQLServer', @v
    exec dbo.dt_getpropertiesbyid_vcs_u @iPropertyObjectId, 'VCSSQLDatabase', @v

if @chObjectType = 'PROC'
begin
    /* Procedure Can have up to three streams
    Drop Stream, Create Stream, GRANT stream */

    exec @iReturn = sp_OACreate @VSSGUID, @iObjectId OUT

    if @iReturn <> 0 GOTO E_OAError

    exec @iReturn = sp_OAMethod @iObjectId,
        N'CheckOut_StoredProcedure',
        NULL,
        @sProjectName = @vchProjectName,
        @sSourceSafeINI = @vchSourceSafeINI,
        @sObjectName = @vchObjectName,
        @sServerName = @vchServerName,
        @sDatabaseName = @vchDatabaseName,
        @sComment = @vchComment,

```

```

                                @sLoginName = @vchLoginName,
                                @sPassword = @vchPassword,
                                @iVCSFlags = @iVCSFlags,
                                @iActionFlag = @iActionFlag

if @iReturn <> 0 GOTO E_OAError

exec @iReturn = sp_OAGetProperty @iObjectId, N'GetStreamObject', @iStreamO

if @iReturn <> 0 GOTO E_OAError

create table #commenttext (id int identity, sourcecode nvarchar(255))

select @vchTempText = N'STUB'
while @vchTempText IS NOT NULL
begin
    exec @iReturn = sp_OAMethod @iStreamObjectId, N'GetStream', @iReturnVa
    if @iReturn <> 0 GOTO E_OAError

    if (@vchTempText IS NOT NULL) insert into #commenttext (sourcecode) se
end

select N'VCS'=sourcecode from #commenttext order by id
select N'SQL'=text from syscomments where id = object_id(@vchObjectName) o

end

CleanUp:
    return

E_OAError:
    exec dbo.dt_displayoaerror_u @iObjectId, @iReturn
    GOTO CleanUp

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

dt_displayoaerror

Summary

Find below summarized details :

- Creation Date : Dec 8 2005 2:53PM

Parameters

Find below input parameters :

1. @iObject : int (4)

2. @iresult : int (4)

Query Analyzer Call example

Find below how to call dt_displayoerror procedure :

```
/* Warning : do not forget to include declare section before to use the call example ! */
```

```
exec dt_displayoerror @iObject, @iresult
```

Source Code

Below find the TSQL statement used to create (or recreate)dt_displayoerror procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[dt_displayoerror]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[dt_displayoerror]

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

CREATE PROCEDURE dbo.dt_displayoerror
    @iObject int,
    @iresult int
as

set nocount on

declare @vchOutput      varchar(255)
declare @hr             int
declare @vchSource      varchar(255)
declare @vchDescription varchar(255)

    exec @hr = sp_OAGetErrorInfo @iObject, @vchSource OUT, @vchDescription OUT

    select @vchOutput= @vchSource + ': ' + @vchDescription
raiserror (@vchOutput,16,-1)

return

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

dt_displayoerror_u

Summary

Find below summarized details :

- Creation Date : Dec 8 2005 2:53PM

Parameters

Find below input parameters :

1. @iObject : int (4)
2. @iresult : int (4)

Query Analyzer Call example

Find below how to call dt_displayoerror_u procedure :

```
/* Warning : do not forget to include declare section before to use the call example ! */
```

```
exec dt_displayoerror_u @iObject, @iresult
```

Source Code

Below find the TSQL statement used to create (or recreate)dt_displayoerror_u procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[dt_displayoerror_u]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[dt_displayoerror_u]

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

CREATE PROCEDURE dbo.dt_displayoerror_u
    @iObject int,
    @iresult int
as

set nocount on

declare @vchOutput      nvarchar(255)
declare @hr             int
declare @vchSource      nvarchar(255)
declare @vchDescription nvarchar(255)

    exec @hr = sp_OAGetErrorInfo @iObject, @vchSource OUT, @vchDescription OUT

select @vchOutput = @vchSource + ': ' + @vchDescription
raiserror (@vchOutput,16,-1)

return
```



```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

dt_droppropertiesbyid

Summary

Find below summarized details :

- Creation Date : Dec 8 2005 2:53PM

Parameters

Find below input parameters :

1. @id : int (4)
2. @property : varchar (64)

Query Analyzer Call example

Find below how to call dt_droppropertiesbyid procedure :

```
/* Warning : do not forget to include declare section before to use the call example ! */
```

```
exec dt_droppropertiesbyid @id, @property
```

Source Code

Below find the TSQL statement used to create (or recreate)dt_droppropertiesbyid procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[dt_droppropertiesbyid]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[dt_droppropertiesbyid]
```

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO
```

```
/*
** Drop one or all the associated properties of an object or an attribute
**
```

```
** dt_dropproperties objid, null or '' -- drop all properties of the object itself
** dt_dropproperties objid, property -- drop the property
*/
create procedure dbo.dt_droppropertiesbyid
    @id int,
    @property varchar(64)
as
    set nocount on

    if (@property is null) or (@property = '')
        delete from dbo.dtproperties where objectid=@id
    else
        delete from dbo.dtproperties
            where objectid=@id and property=@property

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

dt_dropuserobjectbyid

Summary

Find below summarized details :

- Creation Date : Dec 8 2005 2:53PM

Parameters

Find below input parameters :

1. @id : int (4)

Query Analyzer Call example

Find below how to call dt_dropuserobjectbyid procedure :

```
/* Warning : do not forget to include declare section before to use the call example ! */
```

```
exec dt_dropuserobjectbyid @id
```

Source Code

Below find the TSQL statement used to create (or recreate)dt_dropuserobjectbyid procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[dt_dropuserobjectbyid]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[dt_dropuserobjectbyid]
```

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

/*
** Drop an object from the dbo.dtproperties table
*/
create procedure dbo.dt_dropuserobjectbyid
    @id int
as
    set nocount on
    delete from dbo.dtproperties where objectid=@id
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

dt_generateansiname

Summary

Find below summarized details :

- Creation Date : Dec 8 2005 2:53PM

Parameters

Find below input parameters :

1. @name : varchar (255)

Query Analyzer Call example

Find below how to call dt_generateansiname procedure :

```
/* Warning : do not forget to include declare section before to use the call example ! */
```

```
exec dt_generateansiname @name
```

Source Code

Below find the TSQL statement used to create (or recreate)dt_generateansiname procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[dt_generateansiname]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
```

```
drop procedure [dbo].[dt_generateansiname]

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

/*
** Generate an ansi name that is unique in the dtproperties.value column
*/
create procedure dbo.dt_generateansiname(@name varchar(255) output)
as
    declare @prologue varchar(20)
    declare @indexstring varchar(20)
    declare @index integer

    set @prologue = 'MSDT-A-'
    set @index = 1

    while 1 = 1
    begin
        set @indexstring = cast(@index as varchar(20))
        set @name = @prologue + @indexstring
        if not exists (select value from dtproperties where value = @name)
            break

        set @index = @index + 1

        if (@index = 10000)
            goto TooMany
    end

Leave:
    return

TooMany:
    set @name = 'DIAGRAM'
    goto Leave

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

dt_getobjwithprop

Summary

Find below summarized details :

- Creation Date : Dec 8 2005 2:53PM

Parameters

Find below input parameters :

1. @property : varchar (30)
2. @value : varchar (255)

Query Analyzer Call example

Find below how to call dt_getobjwithprop procedure :

```
/* Warning : do not forget to include declare section before to use the call example ! */
```

```
exec dt_getobjwithprop @property, @value
```

Source Code

Below find the TSQL statement used to create (or recreate)dt_getobjwithprop procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[dt_getobjwithprop]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[dt_getobjwithprop]

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

/*
** Retrieve the owner object(s) of a given property
*/
create procedure dbo.dt_getobjwithprop
    @property varchar(30),
    @value varchar(255)
as
    set nocount on

    if (@property is null) or (@property = '')
    begin
        raiserror('Must specify a property name.',-1,-1)
        return (1)
    end

    if (@value is null)
        select objectid id from dbo.dtproperties
            where property=@property
    else
        select objectid id from dbo.dtproperties
            where property=@property and value=@value
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

dt_getobjwithprop_u

Summary

Find below summarized details :

- Creation Date : Dec 8 2005 2:53PM

Parameters

Find below input parameters :

1. @property : varchar (30)
2. @uvalue : nvarchar (510)

Query Analyzer Call example

Find below how to call dt_getobjwithprop_u procedure :

```
/* Warning : do not forget to include declare section before to use the call example ! */
```

```
exec dt_getobjwithprop_u @property, @uvalue
```

Source Code

Below find the TSQL statement used to create (or recreate)dt_getobjwithprop_u procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[dt_getobjwithprop_u]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[dt_getobjwithprop_u]

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

/*
** Retrieve the owner object(s) of a given property
*/
create procedure dbo.dt_getobjwithprop_u
    @property varchar(30),
    @uvalue nvarchar(255)
as
    set nocount on
```

```
if (@property is null) or (@property = '')
begin
    raiserror('Must specify a property name.',-1,-1)
    return (1)
end

if (@uvalue is null)
    select objectid id from dbo.dtproperties
        where property=@property

else
    select objectid id from dbo.dtproperties
        where property=@property and uvalue=@uvalue

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

dt_getpropertiesbyid

Summary

Find below summarized details :

- Creation Date : Dec 8 2005 2:53PM

Parameters

Find below input parameters :

1. @id : int (4)
2. @property : varchar (64)

Query Analyzer Call example

Find below how to call dt_getpropertiesbyid procedure :

```
/* Warning : do not forget to include declare section before to use the call example ! */
```

```
exec dt_getpropertiesbyid @id, @property
```

Source Code

Below find the TSQL statement used to create (or recreate)dt_getpropertiesbyid procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[dt_getpropertiesbyid]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
```

```
drop procedure [dbo].[dt_getpropertiesbyid]

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

/*
** Retrieve properties by id's
**
** dt_getproperties objid, null or '' -- retrieve all properties of the object its
** dt_getproperties objid, property -- retrieve the property specified
*/
create procedure dbo.dt_getpropertiesbyid
    @id int,
    @property varchar(64)
as
    set nocount on

    if (@property is null) or (@property = '')
        select property, version, value, lvalue
            from dbo.dtproperties
            where @id=objectid
    else
        select property, version, value, lvalue
            from dbo.dtproperties
            where @id=objectid and @property=property

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

dt_getpropertiesbyid_u

Summary

Find below summarized details :

- Creation Date : Dec 8 2005 2:53PM

Parameters

Find below input parameters :

1. @id : int (4)
2. @property : varchar (64)

Query Analyzer Call example

Find below how to call dt_getpropertiesbyid_u procedure :


```
/* Warning : do not forget to include declare section before to use the call example ! */
```

```
exec dt_getpropertiesbyid_u @id, @property
```

Source Code

Below find the TSQL statement used to create (or recreate)dt_getpropertiesbyid_u procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[dt_getpropertiesbyid_u]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[dt_getpropertiesbyid_u]

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

/*
** Retrieve properties by id's
**
** dt_getproperties objid, null or '' -- retrieve all properties of the object its
** dt_getproperties objid, property -- retrieve the property specified
*/
create procedure dbo.dt_getpropertiesbyid_u
    @id int,
    @property varchar(64)
as
    set nocount on

    if (@property is null) or (@property = '')
        select property, version, uvalue, lvalue
        from dbo.dtproperties
        where @id=objectid
    else
        select property, version, uvalue, lvalue
        from dbo.dtproperties
        where @id=objectid and @property=property

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

dt_getpropertiesbyid_vcs

Summary

Find below summarized details :

- Creation Date : Dec 8 2005 2:53PM

Parameters

Find below input parameters :

1. @id : int (4)
2. @property : varchar (64)
3. @value : varchar (255)

Query Analyzer Call example

Find below how to call dt_getpropertiesbyid_vcs procedure :

```
/* Warning : do not forget to include declare section before to use the call example ! */
```

```
exec dt_getpropertiesbyid_vcs @id, @property, @value
```

Source Code

Below find the TSQL statement used to create (or recreate)dt_getpropertiesbyid_vcs procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[dt_getpropertiesbyid_vcs]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[dt_getpropertiesbyid_vcs]

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

create procedure dbo.dt_getpropertiesbyid_vcs
    @id          int,
    @property    varchar(64),
    @value       varchar(255) = NULL OUT
as

    set nocount on

    select @value = (
        select value
            from dbo.dtproperties
            where @id=objectid and @property=property
        )

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

dt_getpropertiesbyid_vcs_u

Summary

Find below summarized details :

- Creation Date : Dec 8 2005 2:53PM

Parameters

Find below input parameters :

1. @id : int (4)
2. @property : varchar (64)
3. @value : nvarchar (510)

Query Analyzer Call example

Find below how to call dt_getpropertiesbyid_vcs_u procedure :

```
/* Warning : do not forget to include declare section before to use the call example ! */
```

```
exec dt_getpropertiesbyid_vcs_u @id, @property, @value
```

Source Code

Below find the TSQL statement used to create (or recreate)dt_getpropertiesbyid_vcs_u procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[dt_getpropertiesbyid_vcs_u]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[dt_getpropertiesbyid_vcs_u]

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

create procedure dbo.dt_getpropertiesbyid_vcs_u
    @id          int,
    @property    varchar(64),
    @value       nvarchar(255) = NULL OUT
as

    set nocount on

    select @value = (
```

```
        select uvalue
           from dbo.dtproperties
          where @id=objectid and @property=property
        )

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

dt_isundersourcecontrol

Summary

Find below summarized details :

- Creation Date : Dec 8 2005 2:53PM

Parameters

Find below input parameters :

1. @vchLoginName : varchar (255)
2. @vchPassword : varchar (255)
3. @iWhoToo : int (4)

Query Analyzer Call example

Find below how to call dt_isundersourcecontrol procedure :

```
/* Warning : do not forget to include declare section before to use the call example ! */
```

```
exec dt_isundersourcecontrol @vchLoginName, @vchPassword, @iWhoToo
```

Source Code

Below find the TSQL statement used to create (or recreate)dt_isundersourcecontrol procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[dt_isundersourcecontrol]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[dt_isundersourcecontrol]

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
```

```

GO

create proc dbo.dt_isundersourcecontrol
    @vchLoginName varchar(255) = '',
    @vchPassword  varchar(255) = '',
    @iWhoToo      int = 0 /* 0 => Just check project; 1 => get list of objs */
as

set nocount on

declare @iReturn int
declare @iObjectId int
select @iObjectId = 0

declare @VSSGUID varchar(100)
select @VSSGUID = 'SQLVersionControl.VCS_SQL'

declare @iReturnValue int
select @iReturnValue = 0

declare @iStreamObjectId int
select @iStreamObjectId = 0

declare @vchTempText varchar(255)

    declare @iPropertyObjectId int
    select @iPropertyObjectId = (select objectid from dbo.dtproperties where propert

    declare @vchProjectName  varchar(255)
    declare @vchSourceSafeINI varchar(255)
    declare @vchServerName   varchar(255)
    declare @vchDatabaseName varchar(255)
    exec dbo.dt_getpropertiesbyid_vcs @iPropertyObjectId, 'VCSPProject', @vch
    exec dbo.dt_getpropertiesbyid_vcs @iPropertyObjectId, 'VCSSourceSafeINI', @vch
    exec dbo.dt_getpropertiesbyid_vcs @iPropertyObjectId, 'VCSSQLServer', @vch
    exec dbo.dt_getpropertiesbyid_vcs @iPropertyObjectId, 'VCSSQLDatabase', @vch

    if (@vchProjectName IS NULL) or (@vchSourceSafeINI IS NULL) or (@vchServerName
    begin
        RAISERROR('Not Under Source Control',16,-1)
        return
    end

    if @iWhoToo = 1
    begin

        /* Get List of Procs in the project */
        exec @iReturn = sp_OACreate @VSSGUID, @iObjectId OUT
        if @iReturn <> 0 GOTO E_OAError

        exec @iReturn = sp_OAMethod @iObjectId,
            'GetListOfObjects',
            NULL,
            @vchProjectName,
            @vchSourceSafeINI,
            @vchServerName,
            @vchDatabaseName,
            @vchLoginName,
            @vchPassword

        if @iReturn <> 0 GOTO E_OAError
    
```

```
exec @iReturn = sp_OAGetProperty @iObjectId, 'GetStreamObject', @iStreamOb

if @iReturn <> 0 GOTO E_OAError

create table #ObjectList (id int identity, vchObjectlist varchar(255))

select @vchTempText = 'STUB'
while @vchTempText IS NOT NULL
begin
    exec @iReturn = sp_OAMethod @iStreamObjectId, 'GetStream', @iReturnVal
    if @iReturn <> 0 GOTO E_OAError

    if (@vchTempText IS NOT NULL) insert into #ObjectList (vchObjectlist )
end

select vchObjectlist from #ObjectList order by id

end

CleanUp:
    return

E_OAError:
    exec dbo.dt_displayoerror @iObjectId, @iReturn
    goto CleanUp

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

dt_isundersourcecontrol_u

Summary

Find below summarized details :

- Creation Date : Dec 8 2005 2:53PM

Parameters

Find below input parameters :

1. @vchLoginName : nvarchar (510)
2. @vchPassword : nvarchar (510)
3. @iWhoToo : int (4)

Query Analyzer Call example

Find below how to call dt_isundersourcecontrol_u procedure :

```
/* Warning : do not forget to include declare section before to use the call example ! */
```

```
exec dt_isundersourcecontrol_u @vchLoginName, @vchPassword, @iWhoToo
```

Source Code

Below find the TSQL statement used to create (or recreate)dt_isundersourcecontrol_u procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[dt_isundersourcecontrol_u]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[dt_isundersourcecontrol_u]

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

create proc dbo.dt_isundersourcecontrol_u
    @vchLoginName nvarchar(255) = '',
    @vchPassword nvarchar(255) = '',
    @iWhoToo int = 0 /* 0 => Just check project; 1 => get list of objs */
as

    set nocount on

    declare @iReturn int
    declare @iObjectId int
    select @iObjectId = 0

    declare @VSSGUID nvarchar(100)
    select @VSSGUID = N'SQLVersionControl.VCS_SQL'

    declare @iReturnValue int
    select @iReturnValue = 0

    declare @iStreamObjectId int
    select @iStreamObjectId = 0

    declare @vchTempText nvarchar(255)

    declare @iPropertyObjectId int
    select @iPropertyObjectId = (select objectid from dbo.dtproperties where prope

    declare @vchProjectName nvarchar(255)
    declare @vchSourceSafeINI nvarchar(255)
    declare @vchServerName nvarchar(255)
    declare @vchDatabaseName nvarchar(255)
    exec dbo.dt_getpropertiesbyid_vcs_u @iPropertyObjectId, 'VCSPROject', @v
    exec dbo.dt_getpropertiesbyid_vcs_u @iPropertyObjectId, 'VCSSourceSafeINI', @v
    exec dbo.dt_getpropertiesbyid_vcs_u @iPropertyObjectId, 'VCSSQLServer', @v
    exec dbo.dt_getpropertiesbyid_vcs_u @iPropertyObjectId, 'VCSSQLDatabase', @v

    if (@vchProjectName IS NULL) or (@vchSourceSafeINI IS NULL) or (@vchServerName
begin
    RAISERROR(N'Not Under Source Control',16,-1)
    return
end
```

```
if @iWhoToo = 1
begin

    /* Get List of Procs in the project */
    exec @iReturn = sp_OACreate @VSSGUID, @iObjectId OUT
    if @iReturn <> 0 GOTO E_OAError

    exec @iReturn = sp_OAMethod @iObjectId,
                                N'GetListOfObjects',
                                NULL,
                                @vchProjectName,
                                @vchSourceSafeINI,
                                @vchServerName,
                                @vchDatabaseName,
                                @vchLoginName,
                                @vchPassword

    if @iReturn <> 0 GOTO E_OAError

    exec @iReturn = sp_OAGetProperty @iObjectId, N'GetStreamObject', @iStreamO

    if @iReturn <> 0 GOTO E_OAError

    create table #ObjectList (id int identity, vchObjectlist nvarchar(255))

    select @vchTempText = N'STUB'
    while @vchTempText IS NOT NULL
    begin
        exec @iReturn = sp_OAMethod @iStreamObjectId, N'GetStream', @iReturnVa
        if @iReturn <> 0 GOTO E_OAError

        if (@vchTempText IS NOT NULL) insert into #ObjectList (vchObjectlist )
        end

    select vchObjectlist from #ObjectList order by id
    end

CleanUp:
    return

E_OAError:
    exec dbo.dt_displayoerror_u @iObjectId, @iReturn
    goto CleanUp

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

dt_removefromsourcecontrol

Summary

Find below summarized details :

- Creation Date : Dec 8 2005 2:53PM

Parameters

No Input Parameters Required

Query Analyzer Call example

Find below how to call dt_removefromsourcecontrol procedure :

```
exec dt_removefromsourcecontrol
```

Source Code

Below find the TSQL statement used to create (or recreate)dt_removefromsourcecontrol procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[dt_removefromsourcecontrol]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[dt_removefromsourcecontrol]

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

create procedure dbo.dt_removefromsourcecontrol
as

    set nocount on

    declare @iPropertyObjectId int
    select @iPropertyObjectId = (select objectid from dbo.dtproperties where prope

    exec dbo.dt_droppropertiesbyid @iPropertyObjectId, null

    /* -1 is returned by dt_droppropertiesbyid */
    if @@error <> 0 and @@error <> -1 return 1

    return 0

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

dt_setpropertybyid

Summary

Find below summarized details :

- Creation Date : Dec 8 2005 2:53PM

Parameters

Find below input parameters :

1. @id : int (4)
2. @property : varchar (64)
3. @value : varchar (255)
4. @lvalue : image (16)

Query Analyzer Call example

Find below how to call dt_setpropertybyid procedure :

```
/* Warning : do not forget to include declare section before to use the call example ! */
```

```
exec dt_setpropertybyid @id, @property, @value, @lvalue
```

Source Code

Below find the TSQL statement used to create (or recreate)dt_setpropertybyid procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[dt_setpropertybyid]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[dt_setpropertybyid]

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

/*
** If the property already exists, reset the value; otherwise add property
**      id -- the id in sysobjects of the object
**      property -- the name of the property
**      value -- the text value of the property
**      lvalue -- the binary value of the property (image)
*/
create procedure dbo.dt_setpropertybyid
    @id int,
    @property varchar(64),
    @value varchar(255),
    @lvalue image
as
    set nocount on
    declare @uvalue nvarchar(255)
    set @uvalue = convert(nvarchar(255), @value)
    if exists (select * from dbo.dtproperties
```

```
                where objectid=@id and property=@property)
begin
    --
    -- bump the version count for this row as we update it
    --
    update dbo.dtproperties set value=@value, uvalue=@uvalue, lvalue=@
        where objectid=@id and property=@property
end
else
begin
    --
    -- version count is auto-set to 0 on initial insert
    --
    insert dbo.dtproperties (property, objectid, value, uvalue, lvalue
        values (@property, @id, @value, @uvalue, @lvalue)
end

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

dt_setpropertybyid_u

Summary

Find below summarized details :

- Creation Date : Dec 8 2005 2:53PM

Parameters

Find below input parameters :

1. @id : int (4)
2. @property : varchar (64)
3. @uvalue : nvarchar (510)
4. @lvalue : image (16)

Query Analyzer Call example

Find below how to call dt_setpropertybyid_u procedure :

```
/* Warning : do not forget to include declare section before to use the call example ! */
```

```
exec dt_setpropertybyid_u @id, @property, @uvalue, @lvalue
```

Source Code

Below find the TSQL statement used to create (or recreate)dt_setpropertybyid_u procedure :

```

if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[dt_setpropertybyid_u]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[dt_setpropertybyid_u]

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

/*
** If the property already exists, reset the value; otherwise add property
**     id -- the id in sysobjects of the object
**     property -- the name of the property
**     uvalue -- the text value of the property
**     lvalue -- the binary value of the property (image)
*/
create procedure dbo.dt_setpropertybyid_u
    @id int,
    @property varchar(64),
    @uvalue nvarchar(255),
    @lvalue image
as
    set nocount on
    --
    -- If we are writing the name property, find the ansi equivalent.
    -- If there is no lossless translation, generate an ansi name.
    --
    declare @avalue varchar(255)
    set @avalue = null
    if (@uvalue is not null)
    begin
        if (convert(nvarchar(255), convert(varchar(255), @uvalue)) = @uvalue)
        begin
            set @avalue = convert(varchar(255), @uvalue)
        end
        else
        begin
            if 'DtgSchemaNAME' = @property
            begin
                exec dbo.dt_generateansiname @avalue output
            end
        end
    end
    if exists (select * from dbo.dtproperties
              where objectid=@id and property=@property)
    begin
        --
        -- bump the version count for this row as we update it
        --
        update dbo.dtproperties set value=@avalue, uvalue=@uvalue, lvalue=@lvalue
        where objectid=@id and property=@property
    end
    else
    begin
        --
        -- version count is auto-set to 0 on initial insert
        --

```

```
        insert dbo.dtproperties (property, objectid, value, uvalue, lvalue
                                values (@property, @id, @avalue, @uvalue, @lvalue)
    end
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

dt_validateloginparams

Summary

Find below summarized details :

- Creation Date : Dec 8 2005 2:53PM

Parameters

Find below input parameters :

1. @vchLoginName : varchar (255)
2. @vchPassword : varchar (255)

Query Analyzer Call example

Find below how to call dt_validateloginparams procedure :

```
/* Warning : do not forget to include declare section before to use the call example !*/
```

```
exec dt_validateloginparams @vchLoginName, @vchPassword
```

Source Code

Below find the TSQL statement used to create (or recreate)dt_validateloginparams procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[dt_validateloginparams]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[dt_validateloginparams]

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

create proc dbo.dt_validateloginparams
```

```
        @vchLoginName  varchar(255),
        @vchPassword   varchar(255)
as

set nocount on

declare @iReturn int
declare @iObjectId int
select @iObjectId =0

declare @VSSGUID varchar(100)
select @VSSGUID = 'SQLVersionControl.VCS_SQL'

        declare @iPropertyObjectId int
        select @iPropertyObjectId = (select objectid from dbo.dtproperties where prope

        declare @vchSourceSafeINI varchar(255)
        exec dbo.dt_getpropertiesbyid_vcs @iPropertyObjectId, 'VCSSourceSafeINI', @vch

        exec @iReturn = sp_OACreate @VSSGUID, @iObjectId OUT
        if @iReturn <> 0 GOTO E_OAError

        exec @iReturn = sp_OAMethod @iObjectId,
                                'ValidateLoginParams',
                                NULL,
                                @sSourceSafeINI = @vchSourceSafeINI,
                                @sLoginName = @vchLoginName,
                                @sPassword = @vchPassword
        if @iReturn <> 0 GOTO E_OAError

CleanUp:
        return

E_OAError:
        exec dbo.dt_displayoaerror @iObjectId, @iReturn
        GOTO CleanUp

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

dt_validateloginparams_u

Summary

Find below summarized details :

- Creation Date : Dec 8 2005 2:53PM

Parameters

Find below input parameters :

1. @vchLoginName : nvarchar (510)
2. @vchPassword : nvarchar (510)

Query Analyzer Call example

Find below how to call dt_validateloginparams_u procedure :

```
/* Warning : do not forget to include declare section before to use the call example ! */
```

```
exec dt_validateloginparams_u @vchLoginName, @vchPassword
```

Source Code

Below find the TSQL statement used to create (or recreate)dt_validateloginparams_u procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[dt_validateloginparams_u]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[dt_validateloginparams_u]

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

create proc dbo.dt_validateloginparams_u
    @vchLoginName nvarchar(255),
    @vchPassword nvarchar(255)
as

set nocount on

declare @iReturn int
declare @iObjectId int
select @iObjectId =0

declare @VSSGUID nvarchar(100)
select @VSSGUID = N'SQLVersionControl.VCS_SQL'

declare @iPropertyObjectId int
select @iPropertyObjectId = (select objectid from dbo.dtproperties where prope

declare @vchSourceSafeINI nvarchar(255)
exec dbo.dt_getpropertiesbyid_vcs_u @iPropertyObjectId, 'VCSSourceSafeINI', @v

exec @iReturn = sp_OACreate @VSSGUID, @iObjectId OUT
if @iReturn <> 0 GOTO E_OAError

exec @iReturn = sp_OAMethod @iObjectId,
    N'ValidateLoginParams',
    NULL,
    @sSourceSafeINI = @vchSourceSafeINI,
    @sLoginName = @vchLoginName,
    @sPassword = @vchPassword
if @iReturn <> 0 GOTO E_OAError
```

```
CleanUp:
    return

E_OAError:
    exec dbo.dt_displayoaerror_u @iObjectId, @iReturn
    GOTO CleanUp

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

dt_vcsetEnabled

Summary

Find below summarized details :

- Creation Date : Dec 8 2005 2:53PM

Parameters

No Input Parameters Required

Query Analyzer Call example

Find below how to call dt_vcsetEnabled procedure :

```
exec dt_vcsetEnabled
```

Source Code

Below find the TSQL statement used to create (or recreate)dt_vcsetEnabled procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[dt_vcsetEnabled]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[dt_vcsetEnabled]
```

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO
```

```
create proc dbo.dt_vcsetEnabled
```

```
as
```

```
set nocount on
```



```
declare @iObjectId int
select @iObjectId = 0

declare @VSSGUID varchar(100)
select @VSSGUID = 'SQLVersionControl.VCS_SQL'

    declare @iReturn int
    exec @iReturn = sp_OACreate @VSSGUID, @iObjectId OUT
    if @iReturn <> 0 raiserror('', 16, -1) /* Can't Load Helper DLLC */

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

dt_verstamp006

Summary

Find below summarized details :

- Creation Date : Dec 8 2005 2:53PM

Parameters

No Input Parameters Required

Query Analyzer Call example

Find below how to call dt_verstamp006 procedure :

```
exec dt_verstamp006
```

Source Code

Below find the TSQL statement used to create (or recreate)dt_verstamp006 procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[dt_verstamp006]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[dt_verstamp006]
```

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO
```

```
/*
** This procedure returns the version number of the stored
```

```
**      procedures used by the Microsoft Visual Database Tools.
**      Current version is 7.0.00.
*/
create procedure dbo.dt_verstamp006
as
        select 7000
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

dt_whocheckedout

Summary

Find below summarized details :

- Creation Date : Dec 8 2005 2:53PM

Parameters

Find below input parameters :

1. @chObjectType : char (4)
2. @vchObjectName : varchar (255)
3. @vchLoginName : varchar (255)
4. @vchPassword : varchar (255)

Query Analyzer Call example

Find below how to call dt_whocheckedout procedure :

```
/* Warning : do not forget to include declare section before to use the call example ! */
```

```
exec dt_whocheckedout @chObjectType, @vchObjectName, @vchLoginName, @vchPassword
```

Source Code

Below find the TSQL statement used to create (or recreate)dt_whocheckedout procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[dt_whocheckedout]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[dt_whocheckedout]
```

```

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

create proc dbo.dt_whocheckedout
    @chObjectType char(4),
    @vchObjectName varchar(255),
    @vchLoginName varchar(255),
    @vchPassword varchar(255)

as

set nocount on

declare @iReturn int
declare @iObjectId int
select @iObjectId =0

declare @VSSGUID varchar(100)
select @VSSGUID = 'SQLVersionControl.VCS_SQL'

    declare @iPropertyObjectId int

    select @iPropertyObjectId = (select objectid from dbo.dtproperties where prop

    declare @vchProjectName varchar(255)
    declare @vchSourceSafeINI varchar(255)
    declare @vchServerName varchar(255)
    declare @vchDatabaseName varchar(255)
    exec dbo.dt_getpropertiesbyid_vcs @iPropertyObjectId, 'VCSProject', @vch
    exec dbo.dt_getpropertiesbyid_vcs @iPropertyObjectId, 'VCSSourceSafeINI', @vch
    exec dbo.dt_getpropertiesbyid_vcs @iPropertyObjectId, 'VCSSQLServer', @vch
    exec dbo.dt_getpropertiesbyid_vcs @iPropertyObjectId, 'VCSSQLDatabase', @vch

if @chObjectType = 'PROC'
begin
    exec @iReturn = sp_OACreate @VSSGUID, @iObjectId OUT

    if @iReturn <> 0 GOTO E_OAError

    declare @vchReturnValue varchar(255)
    select @vchReturnValue = ''

    exec @iReturn = sp_OAMethod @iObjectId,
        'WhoCheckedOut',
        @vchReturnValue OUT,
        @sProjectName = @vchProjectName,
        @sSourceSafeINI = @vchSourceSafeINI,
        @sObjectName = @vchObjectName,
        @sServerName = @vchServerName,
        @sDatabaseName = @vchDatabaseName,
        @sLoginName = @vchLoginName,
        @sPassword = @vchPassword

    if @iReturn <> 0 GOTO E_OAError

    select @vchReturnValue

end

CleanUp:
return

```

```
E_OAError:
    exec dbo.dt_displayoerror @iObjectId, @iReturn
    GOTO CleanUp

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

dt_whocheckedout_u

Summary

Find below summarized details :

- Creation Date : Dec 8 2005 2:53PM

Parameters

Find below input parameters :

1. @chObjectType : char (4)
2. @vchObjectName : nvarchar (510)
3. @vchLoginName : nvarchar (510)
4. @vchPassword : nvarchar (510)

Query Analyzer Call example

Find below how to call dt_whocheckedout_u procedure :

```
/* Warning : do not forget to include declare section before to use the call example ! */
```

```
exec dt_whocheckedout_u @chObjectType, @vchObjectName, @vchLoginName, @vchPassword
```

Source Code

Below find the TSQL statement used to create (or recreate)dt_whocheckedout_u procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[dt_whocheckedout_u]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[dt_whocheckedout_u]
```

```

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

create proc dbo.dt_whocheckedout_u
    @chObjectType char(4),
    @vchObjectName nvarchar(255),
    @vchLoginName nvarchar(255),
    @vchPassword nvarchar(255)

as

set nocount on

declare @iReturn int
declare @iObjectId int
select @iObjectId =0

declare @VSSGUID nvarchar(100)
select @VSSGUID = N'SQLVersionControl.VCS_SQL'

    declare @iPropertyObjectId int

    select @iPropertyObjectId = (select objectid from dbo.dtproperties where prop

    declare @vchProjectName nvarchar(255)
    declare @vchSourceSafeINI nvarchar(255)
    declare @vchServerName nvarchar(255)
    declare @vchDatabaseName nvarchar(255)
    exec dbo.dt_getpropertiesbyid_vcs_u @iPropertyObjectId, 'VCSProject', @v
    exec dbo.dt_getpropertiesbyid_vcs_u @iPropertyObjectId, 'VCSSourceSafeINI', @v
    exec dbo.dt_getpropertiesbyid_vcs_u @iPropertyObjectId, 'VCSSQLServer', @v
    exec dbo.dt_getpropertiesbyid_vcs_u @iPropertyObjectId, 'VCSSQLDatabase', @v

    if @chObjectType = 'PROC'
    begin
        exec @iReturn = sp_OACreate @VSSGUID, @iObjectId OUT

        if @iReturn <> 0 GOTO E_OAError

        declare @vchReturnValue nvarchar(255)
        select @vchReturnValue = ''

        exec @iReturn = sp_OAMethod @iObjectId,
            N'WhoCheckedOut',
            @vchReturnValue OUT,
            @sProjectName = @vchProjectName,
            @sSourceSafeINI = @vchSourceSafeINI,
            @sObjectName = @vchObjectName,
            @sServerName = @vchServerName,
            @sDatabaseName = @vchDatabaseName,
            @sLoginName = @vchLoginName,
            @sPassword = @vchPassword

        if @iReturn <> 0 GOTO E_OAError

        select @vchReturnValue

    end

CleanUp:
return

```

```
E_OAError:
    exec dbo.dt_displayoerror_u @iObjectId, @iReturn
    GOTO CleanUp
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

sp_dbk_print_book

Summary

Find below summarized details :

- Creation Date : Nov 13 2005 1:42PM

Parameters

No Input Parameters Required

Query Analyzer Call example

Find below how to call sp_dbk_print_book procedure :

```
exec sp_dbk_print_book
```

Source Code

Below find the TSQL statement used to create (or recreate)sp_dbk_print_book procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[sp_dbk_print_book]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[sp_dbk_print_book]
```

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO
```

```
/*
Copyright (C) 2005 Adrien Sales
```

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

You can contact me by email at : prslump at msn dot com or at my job address : Adriens dot Sales at Experian-Scorex dot com, or still visit my own website : <http://adrien.sales.free.fr/>, by msn (mail and instant messaging at the following address : prslump@msn.com), or still by icq : 197582571

To print the GNU licence, you can call the sp_txt_print_gpl stored procedure on the standard Query Analyzer text output.
*/

```
CREATE          procedure sp_dbk_print_book
as
begin
declare @title nvarchar(300)
set @title = 'SQL Server Objects Report'
    -- print the docbookheader
    --exec sp_dbk_print_header
    print '<book>'
    exec sp_dbk_print_bookinfo @title

    -- print SQLServer Reports details
    exec sp_print_dbk_project_info_chapter
    -- print gpl licence manifest
    exec sp_dbk_print_gpl_chapter
    -- print tables chapter
    exec sp_dbk_print_tables_chapter

    -- print views chapter
    exec sp_dbk_print_views_chapter

    -- print procedures chapter
    exec sp_dbk_print_procedures_chapter

    -- print functions chapter
    exec sp_dbk_print_functions_chapter

    -- print types chapter
    exec sp_dbk_print_types_chapter

    -- print filegroups chapter
    exec sp_dbk_print_filegroups_chapter

    -- print users chapter
    exec sp_dbk_print_users_chapter

    -- print Chapter dedicated to scripts
    exec sp_dbk_print_scripts_chapter
```

```
        print '</book>'

end
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

sp_dbk_print_bookinfo

Summary

Find below summarized details :

- Creation Date : Nov 12 2005 1:40PM

Parameters

Find below input parameters :

1. @title : nvarchar (4000)

Query Analyzer Call example

Find below how to call sp_dbk_print_bookinfo procedure :

```
/* Warning : do not forget to include declare section before to use the call example ! */
```

```
exec sp_dbk_print_bookinfo @title
```

Source Code

Below find the TSQL statement used to create (or recreate)sp_dbk_print_bookinfo procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[sp_dbk_print_bookinfo]'))
```



```
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[sp_dbk_print_bookinfo]
```

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO
```

```
/*
Copyright (C) 2005 Adrien Sales
```

```
This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation.
```

```
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.
```

```
You can contact me by email at : prslump at msn dot com or at my job address :
Adriens dot Sales at Experian-Scorex dot com, or still visit my own website :
http://adrien.sales.free.fr/, by msn (mail and instant messaging at the following
address : prslump@msn.com), or still by icq : 197582571
```

```
To print the GNU licence, you can call the sp_txt_print_gpl stored procedure
on the standard Query Analyzer text output.
*/
```

```
-- sp_dbk_print_bookinfo 'toto'
CREATE procedure sp_dbk_print_bookinfo(@title nvarchar(2000))
as
begin
-- Local Server variables
declare @local_servername varchar(30)
declare @local_servicename varchar(30)
declare @local_dbname varchar(30)
declare @user_name varchar(30)

set @local_servername = cast(@@SERVERNAME as varchar)
set @local_servicename = cast(@@SERVICENAME as varchar)
set @local_dbname = cast(db_name() as varchar)
set @user_name = user_name()

print '<bookinfo>'
print '<title>SQL Server DB (' + @local_dbname + ') Report : ' + isnull(@t
print '<subtitle>'
print '<para>Report built on ' + cast(getdate() as varchar) + ' (local tim
print '<para>Server : ' + @local_servername + '</para>'
print '<para>Service Name : ' + @local_servicename + '</para>'
print '<para>Database Name : ' + @local_dbname + '</para>'
```

```
print '</subtitle>'
print '<authorgroup>'
print '<author><firstname>' + @user_name + '</firstname><surname>DB user<'
print '<author><firstname>Adrien</firstname><surname>Sales</surname><affil'
print '</authorgroup>'
print '</bookinfo>'

end
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

sp_dbk_print_extended_procedure_section

Summary

Find below summarized details :

- Creation Date : Dec 3 2005 5:41PM

Parameters

Find below input parameters :

1. @proc_name : nvarchar (100)

Query Analyzer Call example

Find below how to call sp_dbk_print_extended_procedure_section procedure :

```
/* Warning : do not forget to include declare section before to use the call example ! */
```

```
exec sp_dbk_print_extended_procedure_section @proc_name
```

Source Code

Below find the TSQL statement used to create (or recreate)sp_dbk_print_extended_procedure_section procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[sp_dbk_print_extended_procedure_section]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[sp_dbk_print_extended_procedure_section]

SET QUOTED_IDENTIFIER ON
GO
```

```
SET ANSI_NULLS ON
GO
```

```
/*
Copyright (C) 2005 Adrien Sales
```

```
This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation.
```

```
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.
```

```
You can contact me by email at : prslump at msn dot com or at my job address :
Adriens dot Sales at Experian-Scorex dot com, or still visit my own website :
http://adrien.sales.free.fr/, by msn (mail and instant messaging at the following
address : prslump@msn.com), or still by icq : 197582571
```

```
To print the GNU licence, you can call the sp_txt_print_gpl stored procedure
on the standard Query Analyzer text output.
```

```
*/
CREATE procedure sp_dbk_print_extended_procedure_section (@proc_name nvarchar(50)
as
begin
declare @proc_id int
declare @cr_date datetime
declare @dll_name varchar(50)

if exists(select id from sysobjects where name = @proc_name and xtype = 'X')
begin
select @proc_id = id from sysobjects where name = @proc_name and x
print '<section id = "xp-' + cast(@proc_name as varchar)+ '">'
print '<title>' + @proc_name + '</title>'
print '<para></para>'
-- crdate
-- text from syscomments
select @cr_date = T1.crdate,
       @dll_name = T2.text
from sysobjects T1,
syscomments T2
where T1.id = T2.id
and T1.id = @proc_id
print '<para>'
print '<itemizedlist>'
print '<listitem>Creation date : ' + cast(@cr_date as nvarchar)+ ' '
print '<listitem>' + @dll_name + '</listitem>'
print '</itemizedlist>'
print '</para>'
print '</section>'
end

end

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
```

GO

sp_dbk_print_extended_procedures_chapter

Summary

Find below summarized details :

- Creation Date : Dec 4 2005 4:27PM

Parameters

No Input Parameters Required

Query Analyzer Call example

Find below how to call sp_dbk_print_extended_procedures_chapter procedure :

```
exec sp_dbk_print_extended_procedures_chapter
```

Source Code

Below find the TSQL statement used to create (or recreate)sp_dbk_print_extended_procedures_chapter procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[sp_dbk_print_extended_procedures_chapter]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[sp_dbk_print_extended_procedures_chapter]
```

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO
```

```
/*
Copyright (C) 2005 Adrien Sales
```

```
This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation.
```

```
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.
```

```
You can contact me by email at : prslump at msn dot com or at my job address :
```

Adriens dot Sales at Experian-Scorex dot com, or still visit my own website : <http://adrien.sales.free.fr/>, by msn (mail and instant messaging at the following address : prslump@msn.com), or still by icq : 197582571

To print the GNU licence, you can call the `sp_txt_print_gpl` stored procedure on the standard Query Analyzer text output.

```

*/
CREATE procedure sp_dbk_print_extended_procedures_chapter
as
begin
declare @nb_procs int
declare @proc_name nvarchar(50)
if exists(select id from sysobjects where xtype = 'X')
begin
select @nb_procs = count(*) from sysobjects where xtype = 'X'
print '<chapter id = "xp_procs">'
print '<title>Extended procedures (' + cast(@nb_procs as nvarchar)
declare procs_cursor cursor for select name from sysobjects where
open procs_cursor
fetch next from procs_cursor into @proc_name
while @@fetch_status = 0
begin
exec sp_dbk_print_extended_procedure_section @proc_name
fetch next from procs_cursor into @proc_name
end
close procs_cursor
deallocate procs_cursor
print '</chapter>'
end
end

end

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

```

sp_dbk_print_filegroup_section

Summary

Find below summarized details :

- Creation Date : Nov 13 2005 4:56PM

Parameters

Find below input parameters :

1. @group_name : nvarchar (100)

Query Analyzer Call example

Find below how to call sp_dbk_print_filegroup_section procedure :

```
/* Warning : do not forget to include declare section before to use the call example ! */
```

```
exec sp_dbk_print_filegroup_section @group_name
```

Source Code

Below find the TSQL statement used to create (or recreate)sp_dbk_print_filegroup_section procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[sp_dbk_print_filegroup_section]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[sp_dbk_print_filegroup_section]
```

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO
```

```
/*
Copyright (C) 2005 Adrien Sales
```

```
This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation.
```

```
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.
```

```
You can contact me by email at : prslump at msn dot com or at my job address :
Adriens dot Sales at Experian-Scorex dot com, or still visit my own website :
http://adrien.sales.free.fr/, by msn (mail and instant messaging at the following
address : prslump@msn.com), or still by icq : 197582571
```

```
To print the GNU licence, you can call the sp_txt_print_gpl stored procedure
on the standard Query Analyzer text output.
*/
```

```
-- exec sp_dbk_print_filegroup_section 'PRIMARY'
CREATE procedure sp_dbk_print_filegroup_section(@group_name nvarchar(50))
as
begin
declare @groupid int
declare @allocpolicy int
```

```

declare @status          int

declare @l_fileid        int
declare @l_size          int
declare @l_maxsize      int
declare @l_growth        int
declare @l_status        int
declare @l_perf          int
declare @l_name          nvarchar(100)
declare @l_filename      nvarchar(500)

if exists(select groupid from sysfilegroups)
begin
-- select * from sysfilegroups
select @groupid = groupid,
       @allocpolicy = allocpolicy,
       @status = status
from sysfilegroups
print '<section id = "filegroup-' + cast(@groupid as nvarchar)+ '">'
print '<title>' + @group_name + ' filegroup</title>'
print '<para>Details about ' + @group_name + ' filegroup</para>'

-- summary
print '<section id = "filegroup-' + cast(@groupid as nvarchar)+ '-summary">'
print '<title>Summary</title>'
print '<para>Below, find summarized details of ' + @group_name + ' filegroup : '
print '<itemizedlist>'
print '<listitem>Total size : ' + 'TODO' + '</listitem>'
print '<listitem></listitem>'
print '</itemizedlist>'
print '</para>'
print '</section>'

-- files details
print '<section id = "filegroup-' + cast(@groupid as nvarchar)+ '-files">'
print '<title>Data Files</title>'

-- print a section for each file
declare files_cursor cursor for
select T1.fileid,
       T1.size,
       T1.maxsize,
       T1.growth,
       T1.status,
       T1.perf,
       T1.name,
       rtrim(T2.filename)
from sysfiles T1, sysfiles1 T2
where T1.groupid = @groupid
and T1.fileid = T2.fileid
open files_cursor
fetch next from files_cursor
into @l_fileid,
     @l_size,
     @l_maxsize,
     @l_growth,
     @l_status,
     @l_perf,
     @l_name,
     @l_filename
while @@fetch_status = 0
begin
    print '<section id = "filegroup-' + cast(@groupid as nvarchar)+ '-files-'

```

```
print '<title>' + rtrim(cast(@l_name as nvarchar)) + ' File</title>'
print '<para>Find below summarized details of this file : '
print '<itemizedlist>'
print '<listitem>File name : ' + @l_filename + '</listitem>'
print '<listitem>Max size : ' + cast(@l_maxsize as nvarchar) + '</listitem>'
print '<listitem>Growth : ' + cast(@l_growth as nvarchar) + '</listitem>'
print '<listitem>Status : ' + cast(@l_status as nvarchar) + '</listitem>'
print '<listitem>Perf : ' + cast(@l_perf as nvarchar) + '</listitem>'
print '</itemizedlist>'
print '</para>'

print '</section>'
fetch next from files_cursor
into @l_fileid,
    @l_size,
    @l_maxsize,
    @l_growth,
    @l_status,
    @l_perf,
    @l_name,
    @l_filename

end
close files_cursor
deallocate files_cursor
print '</section>'-- end of files section

print '</section>'

end

end

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

sp_dbk_print_filegroups_book

Summary

Find below summarized details :

- Creation Date : Nov 13 2005 5:53PM

Parameters

No Input Parameters Required

Query Analyzer Call example

Find below how to call sp_dbk_print_filegroups_book procedure :

```
exec sp_dbk_print_filegroups_book
```

Source Code

Below find the TSQL statement used to create (or recreate)sp_dbk_print_filegroups_book procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[sp_dbk_print_filegroups_book]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[sp_dbk_print_filegroups_book]
```

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO
```

```
/*
Copyright (C) 2005 Adrien Sales
```

```
This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation.
```

```
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.
```

```
You can contact me by email at : prslump at msn dot com or at my job address :
Adriens dot Sales at Experian-Scorex dot com, or still visit my own website :
http://adrien.sales.free.fr/, by msn (mail and instant messaging at the following
address : prslump@msn.com), or still by icq : 197582571
```

```
To print the GNU licence, you can call the sp_txt_print_gpl stored procedure
on the standard Query Analyzer text output.
*/
```

```
CREATE procedure sp_dbk_print_filegroups_book
as
begin
declare @title nvarchar(300)
set @title = 'Filegroups Report'
-- print the docbookheader
--exec sp_dbk_print_header
print '<book>'
exec sp_dbk_print_bookinfo @title
-- print chapter
```

```
        exec sp_dbk_print_filegroups_chapter
        print '</book>'
end
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

sp_dbk_print_filegroups_chapter

Summary

Find below summarized details :

- Creation Date : Nov 13 2005 5:38PM

Parameters

No Input Parameters Required

Query Analyzer Call example

Find below how to call sp_dbk_print_filegroups_chapter procedure :

```
exec sp_dbk_print_filegroups_chapter
```

Source Code

Below find the TSQL statement used to create (or recreate)sp_dbk_print_filegroups_chapter procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[sp_dbk_print_filegroups_chapter]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[sp_dbk_print_filegroups_chapter]
```

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO
```

```
/*
Copyright (C) 2005 Adrien Sales
```

```
This program is free software; you can redistribute it and/or
```

modify it under the terms of the GNU General Public License as published by the Free Software Foundation.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

You can contact me by email at : prslump at msn dot com or at my job address : Adriens dot Sales at Experian-Scorex dot com, or still visit my own website : <http://adrien.sales.free.fr/>, by msn (mail and instant messaging at the following address : prslump@msn.com), or still by icq : 197582571

To print the GNU licence, you can call the sp_txt_print_gpl stored procedure on the standard Query Analyzer text output.
*/

```

CREATE      procedure sp_dbk_print_filegroups_chapter
as
begin
declare @l_groupname nvarchar(50)
declare @nb_filegroups int
-- sp_dbk_print_filegroup_section(@group_name nvarchar(50))
if exists(select groupname from sysfilegroups)
begin
    select @nb_filegroups = count(*) from sysfilegroups
    print '<chapter id = "filegroup">'
    print '<title>Storage ( ' + cast(@nb_filegroups as nvarchar) + ' filegroup
    declare group_cursor cursor for
    select groupname
    from sysfilegroups
    open group_cursor
    fetch next from group_cursor into @l_groupname
    while @@fetch_status = 0
    begin
        exec sp_dbk_print_filegroup_section @l_groupname
        fetch next from group_cursor into @l_groupname
    end

    close group_cursor
    deallocate group_cursor
    print '</chapter>'
end

-- select * from sysfilegroups

end

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON

```

GO

sp_dbk_print_function_section

Summary

Find below summarized details :

- Creation Date : Nov 13 2005 2:02PM

Parameters

Find below input parameters :

1. @function_name : nvarchar (100)

Query Analyzer Call example

Find below how to call sp_dbk_print_function_section procedure :

```
/* Warning : do not forget to include declare section before to use the call example ! */
```

```
exec sp_dbk_print_function_section @function_name
```

Source Code

Below find the TSQL statement used to create (or recreate)sp_dbk_print_function_section procedure :

```
if exists(  
select * from dbo.sysobjects  
where id = object_id(N'[dbo].[sp_dbk_print_function_section]')  
and OBJECTPROPERTY(id, N'IsProcedure') = 1)  
drop procedure [dbo].[sp_dbk_print_function_section]
```

```
SET QUOTED_IDENTIFIER ON  
GO  
SET ANSI_NULLS ON  
GO
```

```
/*  
Copyright (C) 2005 Adrien Sales
```

```
This program is free software; you can redistribute it and/or  
modify it under the terms of the GNU General Public License  
as published by the Free Software Foundation.
```

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

You can contact me by email at : prslump at msn dot com or at my job address : Adriens dot Sales at Experian-Scorex dot com, or still visit my own website : <http://adrien.sales.free.fr/>, by msn (mail and instant messaging at the following address : prslump@msn.com), or still by icq : 197582571

To print the GNU licence, you can call the sp_txt_print_gpl stored procedure on the standard Query Analyzer text output.

*/

```

CREATE          procedure sp_dbk_print_function_section(@function_name nvarchar(50))
as
begin
declare @function_id int
declare @creation_date datetime
declare @nb_parameters int

declare @l_parameter_name nvarchar(50)
declare @l_parameter_xtype int
declare @l_parameter_colid nvarchar(50)
declare @l_parameter_length int
declare @l_parameter_xprec int
declare @l_parameter_collation sysname

if exists(select id from sysobjects where name = @function_name and xtype = 'FN')
begin
    select @function_id = id, @creation_date = crdate from sysobjects where name = @function_name
    select @nb_parameters = count(*) from syscolumns where id = @function_id and xtype = 'FN'
    print '<section id = "function-' + cast(@function_id as nvarchar) + '>'
    print '<title>' + @function_name + '</title>'
    -- summary section
    print '<section id = "function-' + cast(@function_id as nvarchar) + '-summary'
    print '<title>Summary</title>'
    print '<para>Find below summarized details : '
    print '<itemizedlist>'
    print '<listitem>Creation Date : ' + cast(@creation_date as nvarchar) + '<'
    print '</itemizedlist>'
    print '</para>'
    print '</section>'-- end of function summary

    -- input parameters section
    print '<section id = "function-' + cast(@function_id as nvarchar) + '-parameters'
    print '<title>Input Parameters (' + cast(@nb_parameters as nvarchar) + ')</'
    if (@nb_parameters = 0)
    begin
        print '<para>No input parameters.</para>'
    end
    else
    begin
        print '<para>Find below input parameters details : '
        print '<itemizedlist>'
        declare parameters_cursor cursor for
            select name,
                   xtype,

```

```

        colid,
        length,
        xprec,
        collation
    from syscolumns
    where id = @function_id
        and name <> ''
    order by colid asc
open parameters_cursor
fetch next from parameters_cursor
into @l_parameter_name,
    @l_parameter_xtype,
    @l_parameter_colid,
    @l_parameter_length,
    @l_parameter_xprec,
    @l_parameter_collation
while @@fetch_status = 0
begin
    print '<listitem>' + cast(@l_parameter_name as nvarchar) +
    fetch next from parameters_cursor
    into @l_parameter_name,
        @l_parameter_xtype,
        @l_parameter_colid,
        @l_parameter_length,
        @l_parameter_xprec,
        @l_parameter_collation
end
close parameters_cursor
deallocate parameters_cursor
print '</itemizedlist>'
print '</para>'
end
print '</section>'-- end of input parameters section

-- output parameter section
select @l_parameter_name = name,
    @l_parameter_xtype = xtype,
    @l_parameter_colid = colid,
    @l_parameter_length = length,
    @l_parameter_xprec = xprec,
    @l_parameter_collation = collation
from syscolumns
where id = @function_id
and name = ''

print '<section id = "function-' + cast(@function_id as nvarchar)+ '-output'
print '<title>Output Parameter</title>'

print '<para>Find below details about the output parameter (required for a
print '<itemizedlist>'
print '<listitem>Type : ' + type_name(@l_parameter_xtype) + '</listitem>'
print '<listitem>Length : ' + cast(@l_parameter_length as nvarchar) + '</l
print '<listitem>Precision : ' + cast(@l_parameter_xprec as nvarchar) + '<
print '<listitem>Collation : ' + cast(@l_parameter_collation as nvarchar)

print '</itemizedlist>'
print '</para>'
print '</section>'

-- source code section
print '<section id = "function-' + cast(@function_id as nvarchar)+ '-code"
print '<title>Source Code</title>'

```

```

print '<para>Below find the TSQL statement used to create (or recreate)' +
declare @source nvarchar(255)
declare source_cursor cursor for select dbo.fx_dbk_get_xmlentities(text) f
open source_cursor
print '<programlisting>'
print 'if exists (select * from dbo.sysobjects'
print 'where id = object_id(N''[' + user_name() + '].[' + @function_name +
print 'and xtype in (N''FN'', N''IF'', N''TF''))'
print 'drop function [' + user_name() + '].[' + @function_name + ']'
print 'GO'
print ''
print 'SET QUOTED_IDENTIFIER OFF '
print 'GO'
print 'SET ANSI_NULLS OFF '
print 'GO'
fetch next from source_cursor into @source
while @@fetch_status = 0
begin
    print @source
    fetch next from source_cursor into @source
end
close source_cursor
deallocate source_cursor
print 'GO'
print 'SET QUOTED_IDENTIFIER OFF '
print 'GO'
print 'SET ANSI_NULLS ON'
print 'GO'
print '</programlisting>'
print '</section>'

print '</section>'

end

end

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

```

sp_dbk_print_functions_book

Summary

Find below summarized details :

- Creation Date : Nov 13 2005 5:53PM

Parameters

No Input Parameters Required

Query Analyzer Call example

Find below how to call sp_dbk_print_functions_book procedure :

```
exec sp_dbk_print_functions_book
```

Source Code

Below find the TSQL statement used to create (or recreate)sp_dbk_print_functions_book procedure :

```
if exists(  
select * from dbo.sysobjects  
where id = object_id(N'[dbo].[sp_dbk_print_functions_book]')  
and OBJECTPROPERTY(id, N'IsProcedure') = 1)  
drop procedure [dbo].[sp_dbk_print_functions_book]
```

```
SET QUOTED_IDENTIFIER ON  
GO  
SET ANSI_NULLS ON  
GO
```

```
/*  
Copyright (C) 2005 Adrien Sales
```

```
This program is free software; you can redistribute it and/or  
modify it under the terms of the GNU General Public License  
as published by the Free Software Foundation.
```

```
This program is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License  
along with this program; if not, write to the Free Software  
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.
```

```
You can contact me by email at : prslump at msn dot com or at my job address :  
Adriens dot Sales at Experian-Scorex dot com, or still visit my own website :  
http://adrien.sales.free.fr/, by msn (mail and instant messaging at the following  
address : prslump@msn.com), or still by icq : 197582571
```

```
To print the GNU licence, you can call the sp_txt_print_gpl stored procedure  
on the standard Query Analyzer text output.  
*/
```

```
CREATE procedure sp_dbk_print_functions_book  
as  
begin  
declare @title nvarchar(50)  
  
set @title = 'User Functions Report'
```



```
-- print the docbookheader
--exec sp_dbk_print_header
print '<book>'
exec sp_dbk_print_bookinfo @title
-- print chapter
exec sp_dbk_print_functions_chapter
print '</book>'

end

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

sp_dbk_print_functions_chapter

Summary

Find below summarized details :

- Creation Date : Nov 13 2005 2:41PM

Parameters

No Input Parameters Required

Query Analyzer Call example

Find below how to call sp_dbk_print_functions_chapter procedure :

```
exec sp_dbk_print_functions_chapter
```

Source Code

Below find the TSQL statement used to create (or recreate)sp_dbk_print_functions_chapter procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[sp_dbk_print_functions_chapter]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[sp_dbk_print_functions_chapter]

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO
```

```

/*
Copyright (C) 2005  Adrien Sales

This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

You can contact me by email at : prslump at msn dot com or at my job address :
Adriens dot Sales at Experian-Scorex dot com, or still visit my own website :
http://adrien.sales.free.fr/, by msn (mail and instant messaging at the following
address : prslump@msn.com), or still by icq : 197582571

To print the GNU licence, you can call the sp_txt_print_gpl stored procedure
on the standard Query Analyzer text output.
*/

CREATE procedure sp_dbk_print_functions_chapter
as
begin
declare @nb_functions int
declare @function_name nvarchar(50)

select @nb_functions = count(*) from sysobjects where xtype = 'FN'
if (@nb_functions > 0)
begin
print '<chapter id = "procedure">'
print '<title>Functions (' + cast(@nb_functions as nvarchar) + ' functs.)</title>'
-- fetch functions
declare functions_cursor cursor for
select name from sysobjects
where xtype = 'FN'
open functions_cursor
fetch next from functions_cursor into @function_name
while @@fetch_status = 0
begin
exec sp_dbk_print_function_section @function_name
fetch next from functions_cursor into @function_name
end

close functions_cursor
deallocate functions_cursor

print '</chapter>'

end
end

GO

```

```
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

sp_dbk_print_gpl_book

Summary

Find below summarized details :

- Creation Date : Nov 18 2005 10:17AM

Parameters

No Input Parameters Required

Query Analyzer Call example

Find below how to call sp_dbk_print_gpl_book procedure :

```
exec sp_dbk_print_gpl_book
```

Source Code

Below find the TSQL statement used to create (or recreate)sp_dbk_print_gpl_book procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[sp_dbk_print_gpl_book]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[sp_dbk_print_gpl_book]
```

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO
```

```
/*
Copyright (C) 2005 Adrien Sales
```

```
This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation.
```

```
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.
```

You can contact me by email at : prslump at msn dot com or at my job address : Adriens dot Sales at Experian-Scorex dot com, or still visit my own website : <http://adrien.sales.free.fr/>, by msn (mail and instant messaging at the following address : prslump@msn.com), or still by icq : 197582571

To print the GNU licence, you can call the sp_txt_print_gpl stored procedure on the standard Query Analyzer text output.
*/

```
CREATE procedure sp_dbk_print_gpl_book as
begin
declare @title nvarchar(300)
set @title = 'SQL Server Reports : GPL Manifest'
    -- print the docbookheader
    --exec sp_dbk_print_header
    print '<book>'
    exec sp_dbk_print_bookinfo @title
    exec sp_dbk_print_gpl_chapter
    print '</book>'

end
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

sp_dbk_print_gpl_chapter

Summary

Find below summarized details :

- Creation Date : Nov 14 2005 1:23PM

Parameters

No Input Parameters Required

Query Analyzer Call example

Find below how to call sp_dbk_print_gpl_chapter procedure :

```
exec sp_dbk_print_gpl_chapter
```

Source Code

Below find the TSQL statement used to create (or recreate)sp_dbk_print_gpl_chapter procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[sp_dbk_print_gpl_chapter]'))
```

```
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[sp_dbk_print_gpl_chapter]
```

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO
```

```
/*
Copyright (C) 2005 Adrien Sales
```

```
This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation.
```

```
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.
```

```
You can contact me by email at : prslump at msn dot com or at my job address :
Adriens dot Sales at Experian-Scorex dot com, or still visit my own website :
http://adrien.sales.free.fr/, by msn (mail and instant messaging at the following
address : prslump@msn.com), or still by icq : 197582571
```

```
To print the GNU licence, you can call the sp_txt_print_gpl stored procedure
on the standard Query Analyzer text output.
*/
```

```
CREATE procedure sp_dbk_print_gpl_chapter
as
begin
```

```
print '<chapter id="gpl">'
print ' <title>GNU General Public License</title>'
print ' <section id="gpl-1">'
print ' <title>Preamble</title>'
print ''
print ' <para>The licenses for most software are designed to take away your '
print ' freedom to share and change it. By contrast, the GNU General Public L
print ' intended to guarantee your freedom to share and change '
print ' free software - to make sure the software is free for all its users.
print ' This General Public License applies to most of the Free Software '
print ' Foundation's software and to any other program whose authors commit '
print ' to using it. (Some other Free Software Foundation software is covered
print ' by the GNU Library General Public License instead.) You can apply it
print ' to your programs, too.</para>'
print ''
print ' <para>When we speak of free software, we are referring to freedom, not
print ' Our General Public Licenses are designed to make sure that you have t
print ' freedom to distribute copies of free software (and charge for this '
print ' service if you wish), that you receive source code or can get it if y
print ' want it, that you can change the software or use pieces of it in new
```

```

print '      programs; and that you know you can do these things.</para>'
print '
print ' <para>To protect your rights, we need to make restrictions that forbid
print ' to deny you these rights or to ask you to surrender the rights. These
print ' restrictions translate to certain responsibilities for you if you dis
print ' tribute copies of the software, or if you modify it.</para>'
print '
print ' <para>For example, if you distribute copies of such a program, whether
print ' for a fee, you must give the recipients all the rights that you have.
print ' You must make sure that they, too, receive or can get the source code. An
print ' author must show them these terms so they know their rights.</para>'
print '
print ' <para>We protect your rights with two steps:'
print '   <orderedlist>'
print ' <listitem>'
print '   <para>copyright the software, and</para>'
print ' </listitem>'
print ' <listitem>'
print '   <para>offer you this license which gives you legal permission to copy,
print '   distribute and/or modify the software.</para>'
print ' </listitem>'
print '   </orderedlist>'
print ' </para>'
print '
print ' <para>Also, for each author's protection and ours, we want to make cert
print ' everyone understands that there is no warranty for this free software
print ' the software is modified by someone else and passed on, we want its
print ' recipients to know that what they have is not the original, so that any
print ' problems introduced by others will not reflect on the original author
print ' reputations.</para>'
print '
print ' <para>Finally, any free program is threatened constantly by software pa
print ' We wish to avoid the danger that redistributors of a free program wil
print ' individually obtain patent licenses, in effect making the program
print ' proprietary. To prevent this, we have made it clear that any patent m
print ' licensed for everyone's free use or not licensed at all.</para>'
print '
print ' <para>The precise terms and conditions for copying, distribution and mo
print ' follow.</para>'
print ' </section>'
print ' <section id="gpl-2">'
print '   <title>TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION<
print '   <section id="gpl-2-0">'
print '     <title>Section 0</title>'
print '     <para>This License applies to any program or other work which contain
print ' placed by the copyright holder saying it may be distributed under the term
print ' of this General Public License. The <quote>Program</quote>, below, refers
print ' program or work, and a
print ' <quote>work based on the Program</quote> means either
print ' the Program or any derivative work under copyright law: that is to say, a
print ' work containing the Program or a portion of it, either verbatim or with
print ' modifications and/or translated into another language. (Hereinafter, trans
print ' is included without limitation in the term
print ' <quote>modification</quote>.) Each licensee is addressed as <quote>you</qu
print '
print '     <para>Activities other than copying, distribution and modification ar
print ' this License; they are outside its scope. The act of running the Program i
print ' restricted, and the output from the Program is covered only if its content
print ' constitute a work based on the Program (independent of having been made by
print ' the Program). Whether that is true depends on what the Program does.</para>
print '   </section>'
print '   <section id="gpl-2-1">'
print '     <title>Section 1</title>'
print '     <para>You may copy and distribute verbatim copies of the Program's so

```

```

print ' receive it, in any medium, provided that you conspicuously and appropriate
print ' publish on each copy an appropriate copyright notice and disclaimer of war
print ' keep intact all the notices that refer to this License and to the absence
print ' warranty; and give any other recipients of the Program a copy of this Lice
print ' along with the Program.</para>'
print ''
print '     <para>You may charge a fee for the physical act of transferring a cop
print ' your option offer warranty protection in exchange for a fee.</para>'
print '     </section>'
print '     <section id="gpl-2-2">'
print '         <title>Section 2</title>'
print '         <para>You may modify your copy or copies of the Program or any portio
print ' forming a work based on the Program, and copy and distribute such modifica
print ' or work under the terms of '
print ' <link linkend="gpl-2-1">Section 1</link> above, provided '
print ' that you also meet all of these conditions:'
print ' <orderedlist numeration="loweralpha">'
print '     <listitem>'
print '         <para>You must cause the modified files to carry prominent notices sta
print ' you changed the files and the date of any change.</para>'
print '     </listitem>'
print '     <listitem>'
print '         <para>You must cause any work that you distribute or publish, that in
print ' in part contains or is derived from the Program or any part thereof,
print ' licensed as a whole at no charge to all third parties under the term
print ' this License.</para>'
print '     </listitem>'
print '     <listitem>'
print '         <para>If the modified program normally reads commands interactively wh
print ' must cause it, when started running for such interactive use in the
print ' ordinary way, to print or display an announcement including an appro
print ' copyright notice and a notice that there is no warranty (or else, sa
print ' that you provide a warranty) and that users may redistribute the pro
print ' under these conditions, and telling the user how to view a copy of t
print ' License. (Exception: If the Program itself is interactive but does n
print ' normally print such an announcement, your work based on the P
print ' required to print an announcement.)</para>'
print '     </listitem>'
print ' </orderedlist>'
print '     </para>'
print ''
print '     <para>These requirements apply to the modified work as a whole. If id
print ' of that work are not derived from the Program, and can be reasonably consi
print ' independent and separate works in themselves, then this License, and its t
print ' do not apply to those sections when you distribute them as separate works.
print ' you distribute the same sections as part of a whole which is a work based
print ' Program, the distribution of the whole must be on the terms of this Licens
print ' permissions for other licensees extend to the entire whole, and thus to ea
print ' every part regardless of who wrote it.</para>'
print ''
print '     <para>Thus, it is not the intent of this section to claim rights or c
print ' to work written entirely by you; rather, the intent is to exercise the rig
print ' the distribution of derivative or collective works based on the Program.</
print ''
print '     <para>In addition, mere aggregation of another work not based on the
print ' (or with a work based on the Program) on a volume of a storage or distribu
print ' does not bring the other work under the scope of this License.</para>'
print '     </section>'
print '     <section id="gpl-2-3">'
print '         <title>Section 3</title>'
print '         <para>You may copy and distribute the Program (or a work based on it,
print ' <link linkend="gpl-2-2">Section 2</link> in object code or executable form
print ' <link linkend="gpl-2-1">Sections 1</link> and '
print ' <link linkend="gpl-2-2">2</link> above provided that you also do one of th

```

```
print ' <orderedlist numeration="loweralpha">'
print '   <listitem>'
print '     <para>Accompany it with the complete corresponding machine-readable source code. The source code,
print '       must be distributed under the terms of Sections 1 and 2 above on a medium that is
print '       customarily used for software interchange; or,</para>'
print '   </listitem>'
print '   <listitem>'
print '     <para>Accompany it with a written offer, valid for at least three years, to provide to any
print '       third party, for a charge no more than your cost of physically performing copying and
print '       distribution, a complete machine-readable copy of the corresponding source code, to be
print '       distributed under the terms of Sections 1 and 2 above on a medium that is customarily
print '       used for software interchange; or,</para>'
print '   </listitem>'
print '   <listitem>'
print '     <para>Accompany it with the information you received as to the offer to distribute the
print '       corresponding source code. (This alternative is allowed only for non-commercial
print '       distribution and only if you received the program in object code or executable form
print '       with such an offer, in accord with Subsection b above.)</para>'
print '   </listitem>'
print ' </orderedlist>'
print ' </para>'
print '
print '   <para>The source code for a work means the preferred form of the work for making
print '   to it. For an executable work, complete source code means all the source code for the work
print '   it contains, plus any associated interface definition files, plus the scripts used to
print '   compile and install the executable. However, as a special exception to the above, the
print '   source code distributed need not include anything that is normally distributed (in
print '   binary form) with the major components (compiler, kernel, and so on) of the operating
print '   system on which the executable runs, unless that component itself accompanies the
print '   executable.
print '
print '   <para>If distribution of executable or object code is made by offering access to a
print '   designated place, then offering equivalent access to copy the source code from the same
print '   designated place counts as distribution of the source code, even though third parties are
print '   not required to copy the source along with the object code.</para>'
print ' </section>'
print ' <section id="gpl-2-4">'
print '   <title>Section 4</title>'
print '   <para>You may not copy, modify, sublicense, or distribute the Program under this License.
print '   Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and
print '   will automatically terminate your rights under this License. Those who have received
print '   copies, or rights, from you under this License will have their licenses terminated so long
print '   as such parties remain in full compliance.</para>'
print ' </section>'
print ' <section id="gpl-2-5">'
print '   <title>Section 5</title>'
print '   <para>You are not required to accept this License, since you have not modified it.
print '   However, if you do modify it, you may not use the modified version to propagate copies of
print '   the Program or its derivative works. These actions are prohibited by law if you do not
print '   accept this License. To convey a copy of the modified version, you must indicate clearly
print '   that the Program is modified, and you must indicate the date of the modification. You
print '   may not offer or distribute the Program (or any work based on the Program) under the
print '   License without indicating the date of the modification. You may indicate the date of
print '   the modification by using the date of the first modification to the Program or works based
print '   on it.</para>'
print ' </section>'
print ' <section id="gpl-2-6">'
print '   <title>Section 6</title>'
print '   <para>Each time you redistribute the Program (or any work based on the Program), the
print '   recipient automatically receives a license from the original licensor to copy, distribute
print '   and modify the Program subject to these terms and conditions. You may not impose any
print '   additional conditions on the recipients' exercise of the rights granted herein. You are
print '   not responsible for enforcing compliance by third parties to this License.</para>'
print ' </section>'
print ' <section id="gpl-2-7">'
print '   <title>Section 7</title>'
print '   <para>If, as a consequence of a court judgment or allegation of patent infringement or
print '   for any other reason (not limited to patent issues), conditions are imposed on you (whether
```



```
print ' agreement or otherwise) that contradict the conditions of this License, th
print ' from the conditions of this License. If you cannot distribute so as to sat
print ' your obligations under this License and any other pertinent obligations, t
print ' you may not distribute the Program at all. For example, if a patent licens
print ' royalty-free redistribution of the Program by all those who receive copies
print ' indirectly through you, then the only way you could satisfy both it and th
print ' to refrain entirely from distribution of the Program.</para>'
print ''
print '     <para>If any portion of this section is held invalid or unenforceable
print ' the balance of the section is intended to apply and the section as a whole
print ' in other circumstances.</para>'
print ''
print '     <para>It is not the purpose of this section to induce you to infringe
print ' right claims or to contest validity of any such claims; this section has t
print ' protecting the integrity of the free software distribution system, which i
print ' license practices. Many people have made generous contributions to the wid
print ' distributed through that system in reliance on consistent application of t
print ' to the author/donor to decide if he or she is willing to distribute softwa
print ' system and a licensee cannot impose that choice.</para>'
print ''
print '     <para>This section is intended to make thoroughly clear what is belie
print ' rest of this License.</para>'
print ' </section>'
print ' <section id="gpl-2-8">'
print '   <title>Section 8</title>'
print '   <para>If the distribution and/or use of the Program is restricted in
print ' or by copyrighted interfaces, the original copyright holder who places the
print ' may add an explicit geographical distribution limitation excluding those c
print ' distribution is permitted only in or among countries not thus excluded. In
print ' incorporates the limitation as if written in the body of this License.</pa
print ' </section>'
print ' <section id="gpl-2-9">'
print '   <title>Section 9</title>'
print '   <para>The Free Software Foundation may publish revised and/or new ver
print ' from time to time. Such new versions will be similar in spirit to the pres
print ' in detail to address new problems or concerns.</para>'
print ''
print '     <para>Each version is given a distinguishing version number. If the P
print ' this License which applies to it and <quote>any later version</quote>, you
print ' and conditions either of that version or of any later version published by
print ' Foundation. If the Program does not specify a version number of this Licen
print ' version ever published by the Free Software Foundation.</para>'
print ' </section>'
print ' <section id="gpl-2-10">'
print '   <title>Section 10</title>'
print '   <para>If you wish to incorporate parts of the Program into other free
print ' conditions are different, write to the author to ask for permission. For s
print ' by the Free Software Foundation, write to the Free Software Foundation; we
print ' for this. Our decision will be guided by the two goals of preserving the f
print ' derivatives of our free software and of promoting the sharing and reuse of
print ' </section>'
print ' <section id="gpl-2-11">'
print '   <title>NO WARRANTY Section 11</title>'
print '   <para>BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WAR
print ' PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE C
print ' OTHER PARTIES PROVIDE THE PROGRAM <quote>AS IS</quote> WITHOUT WARRANTY OF
print ' INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY A
print ' PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM
print ' PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, R
print ' </section>'
print ' <section id="gpl-2-12">'
print '   <title>Section 12</title>'
print '   <para>IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN W
print ' ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTE
```

```

print ' FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL D
print ' USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF
print ' INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE
print ' ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED O
print ' DAMAGES.</para>'
print ''
print '     <para>END OF TERMS AND CONDITIONS</para>'
print ' </section>'
print ' </section>'
print ' <section id="gpl-3">'
print '   <title>How to Apply These Terms to Your New Programs</title>'
print '   <para>If you develop a new program, and you want it to be of the greatest
print '     possible use to the public, the best way to achieve this is to make it
print '     free software which everyone can redistribute and change under these
print '
print '   <para>To do so, attach the following notices to the program.  It is safest
print '     to attach them to the start of each source file to most effectively
print '     convey the exclusion of warranty; and each file should have at least
print '     the <quote>copyright</quote> line and a pointer to where the full not
print '
print '   <para><one line to give the program's name and a brief idea of what it
print '     Copyright (C) <year>     <name of author></para>'
print '
print '   <para>This program is free software; you can redistribute it and/or modify
print '     it under the terms of the GNU General Public License as published by
print '     the Free Software Foundation; either version 2 of the License, or
print '     (at your option) any later version.</para>'
print '
print '   <para>This program is distributed in the hope that it will be useful,
print '     but WITHOUT ANY WARRANTY; without even the implied warranty of
print '     MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
print '     GNU General Public License for more details.</para>'
print '
print '   <para>You should have received a copy of the GNU General Public License
print '     along with this program; if not, write to the Free Software
print '     Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-
print '
print '   <para>Also add information on how to contact you by electronic and paper
print '
print '   <para>If the program is interactive, make it output a short notice like
print '     when it starts in an interactive mode:</para>'
print '
print '   <para>Gnomovision version 69, Copyright (C) year name of author'
print '     Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type <quot
print '     This is free software, and you are welcome to redistribute it'
print '     under certain conditions; type <quote>show c</quote> for details.</pa
print '
print '   <para>The hypothetical commands <quote>show w</quote> and <quote>show c
print '     show the appropriate parts of the General Public License.  Of course,
print '     use may be called something other than <quote>show w</quote> and <quo
print '     they could even be mouse-clicks or menu items--whatever suits your pr
print '
print '   <para>You should also get your employer (if you work as a programmer) or
print '     school, if any, to sign a <quote>copyright disclaimer</quote> for the
print '     necessary.  Here is a sample; alter the names:</para>'
print '
print '   <para>Yoyodyne, Inc., hereby disclaims all copyright interest in the pr
print '     <quote>Gnomovision</quote> (which makes passes at compilers) written
print '
print '   <para><signature of Ty Coon>, 1 April 1989'
print '     Ty Coon, President of Vice</para>'
print '
print '   <para>This General Public License does not permit incorporating your pr
print '     proprietary programs.  If your program is a subroutine library, you m

```

```
print '      consider it more useful to permit linking proprietary applications wi
print '      library. If this is what you want to do, use the GNU Library General
print '      Public License instead of this License.</para>'
print ' </section>'
print '</chapter>'
end
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

sp_dbk_print_header

Summary

Find below summarized details :

- Creation Date : Nov 12 2005 1:35PM

Parameters

No Input Parameters Required

Query Analyzer Call example

Find below how to call sp_dbk_print_header procedure :

```
exec sp_dbk_print_header
```

Source Code

Below find the TSQL statement used to create (or recreate)sp_dbk_print_header procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[sp_dbk_print_header]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[sp_dbk_print_header]
```

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO
```

```
/*
Copyright (C) 2005 Adrien Sales
```

```
This program is free software; you can redistribute it and/or
```

modify it under the terms of the GNU General Public License as published by the Free Software Foundation.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

You can contact me by email at : prslump at msn dot com or at my job address : Adriens dot Sales at Experian-Scorex dot com, or still visit my own website : <http://adrien.sales.free.fr/>, by msn (mail and instant messaging at the following address : prslump@msn.com), or still by icq : 197582571

To print the GNU licence, you can call the sp_txt_print_gpl stored procedure on the standard Query Analyzer text output.
*/

```
CREATE procedure sp_dbk_print_header
as
begin
    print '<?xml version="1.0" encoding="UTF-8"?>'
    print '<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook XML V4.2//EN" "http://

end

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

sp_dbk_print_procedure_section

Summary

Find below summarized details :

- Creation Date : Nov 13 2005 11:49AM

Parameters

Find below input parameters :

1. @proc_name : nvarchar (100)

Query Analyzer Call example

Find below how to call sp_dbk_print_procedure_section procedure :

```
/* Warning : do not forget to include declare section before to use the call example ! */
```

```
exec sp_dbk_print_procedure_section @proc_name
```

Source Code

Below find the TSQL statement used to create (or recreate)sp_dbk_print_procedure_section procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[sp_dbk_print_procedure_section]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[sp_dbk_print_procedure_section]
```

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO
```

```
/*
Copyright (C) 2005 Adrien Sales
```

```
This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation.
```

```
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.
```

```
You can contact me by email at : prslump at msn dot com or at my job address :
Adriens dot Sales at Experian-Scorex dot com, or still visit my own website :
http://adrien.sales.free.fr/, by msn (mail and instant messaging at the following
address : prslump@msn.com), or still by icq : 197582571
```

```
To print the GNU licence, you can call the sp_txt_print_gpl stored procedure
on the standard Query Analyzer text output.
```

```
*/
```

```
-- exec sp_dbk_print_procedure_section 'sp_dbk_print_table_book'
CREATE          procedure sp_dbk_print_procedure_section(@proc_name nvarchar(50))
as
```

```

begin
declare @procedure_id int
declare @creation_date datetime

declare @parameter_name          nvarchar(50)
declare @parameter_xtype int
declare @parameter_length int
declare @parameter_xprec int
declare @parameter_colid int
declare @parameter_collation nvarchar(50)
declare @nb_parameters          int
declare @i                      int

set @nb_parameters = 0

if exists(select id from sysobjects where name = @proc_name and xtype = 'P')
begin

    select @procedure_id = id, @creation_date = crdate from sysobjects where name = @proc_name
    select @nb_parameters = count(*) from syscolumns where id = @procedure_id
    print '<section id = "proc-' + cast(@procedure_id as nvarchar)+ '">'
    print '<title>' + @proc_name + '</title>'
    -- summary
    print '<section id = "proc-' + cast(@procedure_id as nvarchar)+ '-summary">'
    print '<title>/'* + @proc_name+ '*/ + 'Summary</title>'
    print '<para>Find below summarized details : '
    print '<itemizedlist>'
    print '<listitem>Creation Date : ' + cast(@creation_date as nvarchar) + ' '
    --print '<listitem></listitem>'
    print '</itemizedlist>'
    print '</para>'
    print '</section>'-- end of storproc summary
    -- procedure parameters
    print '<section id = "proc-' + cast(@procedure_id as nvarchar)+ '-parameters">'
    print '<title>Parameters</title>'
    if exists(select name from syscolumns where id = @procedure_id)
    begin
        -- there is at least one input parameter
        print '<para>Find below input parameters : '
        print '<orderedlist>'
        declare parameters_cursor cursor for
        select name,
               xtype,
               length,
               xprec,
               colid,
               collation
        from syscolumns
        where id = @procedure_id
        order by colid asc
        open parameters_cursor
        fetch next from parameters_cursor into @parameter_name,
        @parameter_xtype,
        @parameter_length,
        @parameter_xprec,
        @parameter_colid,
        @parameter_collation
        while @@fetch_status =0
        begin
            print '<listitem>' + @parameter_name + ' : ' + type_name(@parameter_xtype)
            fetch next from parameters_cursor into @parameter_name,
            @parameter_xtype,

```

```

        @parameter_length,
        @parameter_xprec,
        @parameter_colid,
        @parameter_collation
    end-- end of parameters fetching
    close parameters_cursor
    --deallocate parameters_cursor
    print '</orderedlist>'
    print '</para>'
end
else
begin
    print 'No Input Parameters Required'
end
print '</section>'-- end of parameters section
-- call example section
print '<section id = "proc-' + cast(@procedure_id as nvarchar)+ '-example"'
print '<title>Query Analyzer Call example</title>'
print '<para>Find below how to call ' + @proc_name + ' procedure : </para>'

if exists (select id from syscolumns where id = @procedure_id)
begin
    set @i = 0
    print '<para>'
    print '/* Warning : do not forget to include declare section before'
    print '*/</para>'

    print '<para>exec ' + @proc_name + ''
    -- parse input parameters to build call example
    open parameters_cursor
    fetch next from parameters_cursor into @parameter_name,
        @parameter_xtype,
        @parameter_length,
        @parameter_xprec,
        @parameter_colid,
        @parameter_collation
    if (@nb_parameters = 1)
    begin
        print @parameter_name
    end
    else
    begin
        print @parameter_name + ', '
    end
    while @@fetch_status = 0
    begin
        fetch next from parameters_cursor into @parameter_name,
            @parameter_xtype,
            @parameter_length,
            @parameter_xprec,
            @parameter_colid,
            @parameter_collation
        set @i = @i + 1
        if (@i = @nb_parameters - 1)
        begin
            print @parameter_name
        end
        else if (@i < @nb_parameters -1)
        begin
            print @parameter_name + ', '
        end
    end
end
end
end

```

```

        print '</para>'
        close parameters_cursor
        deallocate parameters_cursor
    end
    else
    begin
        print '<para>exec ' + @proc_name + '</para>'
    end
    print '</section>'-- end of Query analyzer call example section

-- procedure source code
print '<section id = "proc-' + cast(@procedure_id as nvarchar)+ '-code">'
print '<title>Source Code</title>'
print '<para>Below find the TSQL statement used to create (or recreate)' +
declare @source nvarchar(255)
declare source_cursor cursor for select dbo.fx_dbk_get_xmlentities(text) f
open source_cursor
print '<programlisting>'
print 'if exists('
print 'select * from dbo.sysobjects'
print 'where id = object_id(N''[' + user_name()+ '].[' + @proc_name + '])'
print 'and OBJECTPROPERTY(id, N''IsProcedure'') = 1)'
print 'drop procedure [' + user_name()+ '].[' + @proc_name + ']'
print ''
print ''
print 'SET QUOTED_IDENTIFIER ON '
print 'GO'
print 'SET ANSI_NULLS ON '
print 'GO'
print ''
print ''

fetch next from source_cursor into @source
while @@fetch_status = 0
begin
    print @source
    fetch next from source_cursor into @source
end
close source_cursor
deallocate source_cursor
print 'GO'
print 'SET QUOTED_IDENTIFIER OFF '
print 'GO'
print 'SET ANSI_NULLS ON'
print 'GO'
print '</programlisting>'
print '</section>'

print '</section>'
end

end

```

```

GO
SET QUOTED_IDENTIFIER OFF

```



```
GO
SET ANSI_NULLS ON
GO
```

sp_dbk_print_procedures_book

Summary

Find below summarized details :

- Creation Date : Nov 13 2005 12:10PM

Parameters

No Input Parameters Required

Query Analyzer Call example

Find below how to call sp_dbk_print_procedures_book procedure :

```
exec sp_dbk_print_procedures_book
```

Source Code

Below find the TSQL statement used to create (or recreate)sp_dbk_print_procedures_book procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[sp_dbk_print_procedures_book]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[sp_dbk_print_procedures_book]
```

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO
```

```
/*
Copyright (C) 2005 Adrien Sales
```

```
This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation.
```

```
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.
```

You can contact me by email at : prslump at msn dot com or at my job address : Adriens dot Sales at Experian-Scorex dot com, or still visit my own website : <http://adrien.sales.free.fr/>, by msn (mail and instant messaging at the following address : prslump@msn.com), or still by icq : 197582571

To print the GNU licence, you can call the sp_txt_print_gpl stored procedure on the standard Query Analyzer text output.

```
*/  
  
CREATE procedure sp_dbk_print_procedures_book  
as  
begin  
declare @title nvarchar(300)  
set @title = 'User Procedures Report'  
    -- print the docbookheader  
    --exec sp_dbk_print_header  
    print '<book>'  
    exec sp_dbk_print_bookinfo @title  
    -- print chapter  
    exec sp_dbk_print_procedures_chapter  
    print '</book>'  
  
end  
  
GO  
SET QUOTED_IDENTIFIER OFF  
GO  
SET ANSI_NULLS ON  
GO
```

sp_dbk_print_procedures_chapter

Summary

Find below summarized details :

- Creation Date : Nov 13 2005 12:07PM

Parameters

No Input Parameters Required

Query Analyzer Call example

Find below how to call sp_dbk_print_procedures_chapter procedure :

```
exec sp_dbk_print_procedures_chapter
```

Source Code

Below find the TSQL statement used to create (or recreate)sp_dbk_print_procedures_chapter procedure :

```
if exists(  

```

```
select * from dbo.sysobjects
where id = object_id(N'[dbo].[sp_dbk_print_procedures_chapter]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[sp_dbk_print_procedures_chapter]
```

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO
```

```
/*
Copyright (C) 2005 Adrien Sales
```

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

You can contact me by email at : prslump at msn dot com or at my job address : Adriens dot Sales at Experian-Scorex dot com, or still visit my own website : <http://adrien.sales.free.fr/>, by msn (mail and instant messaging at the following address : prslump@msn.com), or still by icq : 197582571

To print the GNU licence, you can call the sp_txt_print_gpl stored procedure on the standard Query Analyzer text output.
*/

```
CREATE      procedure sp_dbk_print_procedures_chapter
as
begin
declare @procedure_name nvarchar(50)
declare @nb_procedures int
```

```
if exists (select id from sysobjects where xtype = 'P')
begin
select @nb_procedures = count(*) from sysobjects where xtype = 'P'
print '<chapter id = "procs">'
print '<title>Procedures (' + cast(@nb_procedures as nvarchar)+ ' procs.)<
-- fetch procedures
declare procedure_cursor cursor for
select name from sysobjects where xtype = 'P' order by name
open procedure_cursor
fetch next from procedure_cursor into @procedure_name
while @@fetch_status = 0
begin
exec sp_dbk_print_procedure_section @procedure_name
fetch next from procedure_cursor into @procedure_name
end
close procedure_cursor
deallocate procedure_cursor
print '</chapter>'
```

```
        end
    end
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

sp_dbk_print_scripts_chapter

Summary

Find below summarized details :

- Creation Date : Dec 8 2005 2:41PM

Parameters

No Input Parameters Required

Query Analyzer Call example

Find below how to call sp_dbk_print_scripts_chapter procedure :

```
exec sp_dbk_print_scripts_chapter
```

Source Code

Below find the TSQL statement used to create (or recreate)sp_dbk_print_scripts_chapter procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[sp_dbk_print_scripts_chapter]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[sp_dbk_print_scripts_chapter]
```

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO
```

```
/*
Copyright (C) 2005 Adrien Sales
```

```
This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation.
```

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

You can contact me by email at : prslump at msn dot com or at my job address : Adriens dot Sales at Experian-Scorex dot com, or still visit my own website : <http://adrien.sales.free.fr/>, by msn (mail and instant messaging at the following address : prslump@msn.com), or still by icq : 197582571

To print the GNU licence, you can call the sp_txt_print_gpl stored procedure on the standard Query Analyzer text output.
*/

```
CREATE procedure sp_dbk_print_scripts_chapter
as
begin
    print '<chapter id = "scripts">'
    print '<title>Management scripts</title>'
    print '<subtitle>A set of scripts to make statistics, diagnose, admin tas
exec sp_dbk_print_scripts_tables_stats
    print '</chapter>'

end

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

sp_dbk_print_scripts_tables_stats

Summary

Find below summarized details :

- Creation Date : Dec 8 2005 2:49PM

Parameters

No Input Parameters Required

Query Analyzer Call example

Find below how to call sp_dbk_print_scripts_tables_stats procedure :

```
exec sp_dbk_print_scripts_tables_stats
```

Source Code

Below find the TSQL statement used to create (or recreate)sp_dbk_print_scripts_tables_stats procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[sp_dbk_print_scripts_tables_stats]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[sp_dbk_print_scripts_tables_stats]

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

/*
Copyright (C) 2005 Adrien Sales

This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

You can contact me by email at : prslump at msn dot com or at my job address :
Adriens dot Sales at Experian-Scorex dot com, or still visit my own website :
http://adrien.sales.free.fr/, by msn (mail and instant messaging at the following
address : prslump@msn.com), or still by icq : 197582571

To print the GNU licence, you can call the sp_txt_print_gpl stored procedure
on the standard Query Analyzer text output.
*/
CREATE procedure sp_dbk_print_scripts_tables_stats
as
begin
print '<section id = "scripts-table_stats">'
print '<title>Tables Statistics Scripts</title>'

print '</section>'

end

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

sp_dbk_print_table_book

Summary

Find below summarized details :

- Creation Date : Nov 12 2005 2:11PM

Parameters

Find below input parameters :

1. @table_name : nvarchar (500)

Query Analyzer Call example

Find below how to call sp_dbk_print_table_book procedure :

```
/* Warning : do not forget to include declare section before to use the call example ! */
```

```
exec sp_dbk_print_table_book @table_name
```

Source Code

Below find the TSQL statement used to create (or recreate)sp_dbk_print_table_book procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[sp_dbk_print_table_book]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[sp_dbk_print_table_book]
```

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO
```

```
/*
Copyright (C) 2005 Adrien Sales
```

```
This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation.
```

```
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.
```

```
You can contact me by email at : prslump at msn dot com or at my job address :
Adriens dot Sales at Experian-Scorex dot com, or still visit my own website :
http://adrien.sales.free.fr/, by msn (mail and instant messaging at the following
```

```
address : prslump@msn.com), or still by icq : 197582571
```

To print the GNU licence, you can call the `sp_txt_print_gpl` stored procedure on the standard Query Analyzer text output.

```
*/

-- exec sp_dbk_print_table_book 'Products'

CREATE    procedure sp_dbk_print_table_book(@table_name nvarchar(250))
as
begin
declare @title nvarchar(300)
set @title = @table_name + ' Table Report'
-- print the docbookheader
--exec sp_dbk_print_header
print '<book>'
exec sp_dbk_print_bookinfo @title
print '<chapter>'
print '<title>Table Report</title>'
exec sp_dbk_print_table_section @table_name
print '</chapter>'
print '</book>'

end

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

sp_dbk_print_table_section

Summary

Find below summarized details :

- Creation Date : Nov 12 2005 1:53PM

Parameters

Find below input parameters :

1. @table_name : nvarchar (500)

Query Analyzer Call example

Find below how to call `sp_dbk_print_table_section` procedure :

```
/* Warning : do not forget to include declare section before to use the call example ! */
```

```
exec sp_dbk_print_table_section @table_name
```


Source Code

Below find the TSQL statement used to create (or recreate)sp_dbk_print_table_section procedure :

```

if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[sp_dbk_print_table_section]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[sp_dbk_print_table_section]

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

/*
Copyright (C) 2005  Adrien Sales

This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

You can contact me by email at : prslump at msn dot com or at my job address :
Adriens dot Sales at Experian-Scorex dot com, or still visit my own website :
http://adrien.sales.free.fr/, by msn (mail and instant messaging at the following
address : prslump@msn.com), or still by icq : 197582571

To print the GNU licence, you can call the sp_txt_print_gpl stored procedure
on the standard Query Analyzer text output.
*/

-- exec sp_dbk_print_table_section 'Products'
CREATE procedure sp_dbk_print_table_section(@table_name nvarchar(250))
as
begin
    -- first get the internal sql server id of the table
    declare @table_id int
    declare @table_creation_date datetime
    declare @table_rowsize int

    declare @l_column_name varchar(40)
    declare @l_column_type_id tinyint
    declare @l_iscomputed int
    declare @l_length int
    declare @l_xprec int
    declare @l_xscale int
    declare @l_isnullable int
    declare @l_filegroup_name nvarchar(50)
    declare @nb_indexes int
    declare @l_index_status int
    declare @l_index_indid int
    declare @l_index_keycnt int

```

```

declare @l_index_groupid int
declare @l_index_rowcnt      int
declare @l_index_xmaxlen int
declare @l_index_maxirow int
declare @l_index_lockflags int
declare @l_index_name      nvarchar(50)
declare @l_index_filegroup nvarchar(50)
declare @l_index_column_name nvarchar(50)
declare @l_index_column_id int
declare @l_index_column_type int
declare @l_index_column_keyno int

if exists( select id, name from sysobjects where xtype = 'u' and name = @table_name)
begin
    select @table_id = id
    from sysobjects
    where xtype = 'u'
      and name = @table_name
    print '<section id = "table-' + cast(@table_id as nvarchar)+ '>'
    print '<title>' + @table_name+ ' Table</title>'
    -- Table summary
    print '<section id = "table-' + cast(@table_id as nvarchar)+ '-summary'
    print '<title>Summary</title>'
    -- high level infos on the table
    print '<para>'
    print 'Summarized details : '

    print '<itemizedlist>'

    -- retrieve filegroup
    select @l_filegroup_name = s.groupname
    from sysfilegroups s,
    sysindexes i
    where i.id = @table_id
      and i.indid < 2
      and i.groupid = s.groupid
    print '<listitem>Filegroup : ' + @l_filegroup_name + '</listitem>'
    --print '<listitem>Type : </listitem>'
    select @table_creation_date = crdate from sysobjects where
    print '<listitem>Creation Date : ' + cast(@table_creation_date as nvarchar)
    select @table_rowsize = sum(length) from syscolumns where
    print '<listitem>(Max)Table Row Size (should not be greater than 8000)'
    print '</itemizedlist>'

    print '</para>'
    print '</section>'-- end of summary section
    print '<section id = "table-' + cast(@table_id as nvarchar)+ '-columns'
    print '<title>Columns</title>'
    -- print column headers
    print '<table frame="all"><title>' + @table_name + ' Columns</title>'
    print '<tgroup cols="10" align="left" colsep="1" rowsep="1">'
    print '<colspec colname="c1"/>'
    print '<colspec colname="c2"/>'
    print '<colspec colname="c3"/>'
    print '<colspec colname="c4"/>'
    print '<colspec colname="c5"/>'
    print '<colspec colname="c6"/>'
    print '<colspec colname="c7"/>'
    print '<colspec colname="c8"/>'
    print '<colspec colname="c9"/>'
    print '<colspec colname="c10"/>'
    print '<spanspec spanname="hspan" namest="c1" nameend="c2">'
    print '<spanspec spanname="hspan" namest="c2" nameend="c3">'
    print '<spanspec spanname="hspan" namest="c3" nameend="c4">'

```

```

print '<spansec spanname="hspan" namest="c4" nameend="c5"'
print '<spansec spanname="hspan" namest="c5" nameend="c6"'
print '<spansec spanname="hspan" namest="c6" nameend="c7"'
print '<spansec spanname="hspan" namest="c7" nameend="c8"'
print '<spansec spanname="hspan" namest="c8" nameend="c9"'
print '<spansec spanname="hspan" namest="c9" nameend="c10"'
print '<thead>'
print '<row>'
print '<entry>Column Name</entry>'
print '  <entry>Type</entry>'
print '  <entry>Computed</entry>'
print '  <entry>Length</entry>'
print '  <entry>Prec</entry>'
print '  <entry>Scale</entry>'
print '  <entry>Nullable</entry>'
print '  <entry>Trim Trailing Blanks</entry>'
print '  <entry>Fixed Length Null</entry>'
print '  <entry>Collation</entry>'
print '</row>'
print '</thead>'
print '<tbody>'
-- display details on columns
declare column_cursor cursor
for select name,
         xtype,
         iscomputed,
         length,
         xprec,
         xscale,
         isnullable
from syscolumns
where id = @table_id
order by colid asc
open column_cursor
fetch next from column_cursor
into @l_column_name,
     @l_column_type_id,
     @l_iscomputed,
     @l_length,
     @l_xprec,
     @l_xscale,
     @l_isnullable
while @@fetch_status = 0
begin
  print '<row>'
  print '  <entry>' + @l_column_name + '</entry>'
  print '  <entry>' + type_name(@l_column_type_id) +
  if (@l_iscomputed = 1)
  begin
    print '    <entry>True</entry>'
  end
  else
  begin
    print '    <entry>False</entry>'
  end
  print '  <entry>' + cast(@l_length as nvarchar) +
  print '  <entry>' + cast(@l_xprec as nvarchar) +
  print '  <entry>' + cast(@l_xscale as nvarchar) +
  if (@l_isnullable = 0)
  begin
    print '    <entry>No</entry>'
  end
  else
  begin

```

```

        print ' <entry>Yes</entry>'
    end
    print ' <entry>NA</entry>'
    print ' <entry>NA</entry>'
    print ' <entry>NA</entry>'
    print '</row>'
    fetch next
    from column_cursor
    into @l_column_name,
        @l_column_type_id,
        @l_iscomputed,
        @l_length,
        @l_xprec,
        @l_xscale,
        @l_isnullable
end
close column_cursor
deallocate column_cursor

    print '</tbody>'
    print '</tgroup>'
    print '</table>'
print '</section>'-- end of columns section
-- indexes section
-- count number of indexes
select @nb_indexes = count(*)
    from sysindexes
    where id = @table_id
    and keys is not null
if (@nb_indexes > 0)
begin
    print '<section id = "table-' + cast(@table_id as nvarchar(10)) + '>'
    print '<title>Indexes and Primary Keys</title>'
    declare indexes_cursor cursor for
    select status,
        indid,
        keycnt,
        groupid,
        rowcnt,
        xmaxlen,
        maxirow,
        lockflags,
        name
    from sysindexes
    where id = @table_id
    and keys is not null
    open indexes_cursor
    fetch next from indexes_cursor into @l_index_status,
        @l_index_indid,
        @l_index_keycnt,
        @l_index_groupid,
        @l_index_rowcnt,
        @l_index_xmaxlen,
        @l_index_maxirow,
        @l_index_lockflags,
        @l_index_name
    while @@fetch_status = 0
    begin
        print '<section id = "table-' + cast(@table_id as nvarchar(10)) + '>'
        print '<title>' + @l_index_name + '</title>'
        -- summary section
        print '<section id = "table-' + cast(@table_id as nvarchar(10)) + '>'
        print '<title>Summary</title>'
    end
end

```

```

print '<para>'
print '<itemizedlist>'
-- retrieve index filegroup
select @l_index_filegroup = groupname
       from sysfilegroups
       where groupid = @l_index_groupid
print '<listitem>Filegroup : ' + @l_index_filegroup+ '</li>'
print '<listitem>Nb Columns : ' + cast(@l_index_keycnt as nvarchar)
print '<listitem>Row count : ' + cast(@l_index_rowcnt as nvarchar)
print '<listitem>Max Len : ' + cast(@l_index_xmaxlen as nvarchar)
print '<listitem>Max irow : ' + cast(@l_index_maxirow as nvarchar)
print '<listitem>Lock Flags : ' + cast(@l_index_lockflags as nvarchar)
print '<listitem>Clustered : </listitem>'
print '</itemizedlist>'
print '</para>'
print '</section>'-- close summary section

-- columns details
print '<section id = "table-' + cast(@table_id as nvarchar) + '>'

print '<title>Columns Details</title>'
print '<para>Below are details on the columns of the ' + @table_name + '
-- build the cursor that will parse columns of the index
declare index_columns_cursor cursor for
       select cols.name,
              keys.colid,
              cols.xtype,
              keys.keyno--,
              --'todo' as sort_order
       from sysindexkeys keys, syscolumns cols
       where keys.id = @table_id
       and keys.indid = @l_index_indid
       and cols.id = @table_id
       and cols.colid = keys.colid
open index_columns_cursor
fetch next from index_columns_cursor
       into @l_index_column_name,
           @l_index_column_id,
           @l_index_column_type,
           @l_index_column_keyno

-- print table header
-- print column headers
print '<table frame="all"><title>Index ' + @l_index_name + '
print '<tgroup cols="5" align="left" colsep="1" rowsep="1"
print '<colspec colname="c1"/>'
print '<colspec colname="c2"/>'
print '<colspec colname="c3"/>'
print '<colspec colname="c4"/>'
print '<colspec colname="c5"/>'
print '<spanspec spanname="hspan" namest="c1" nameend="c2"
print '<spanspec spanname="hspan" namest="c2" nameend="c3"
print '<spanspec spanname="hspan" namest="c3" nameend="c4"
print '<spanspec spanname="hspan" namest="c4" nameend="c5"

print '<thead>'
print '<row>'
print '<entry>Column Name</entry>'
print '<entry>Type</entry>'
print '<entry>Column Id</entry>'
print '<entry>Key Id</entry>'
print '<entry>Asc Sorted ?</entry>'
print '</row>'
print '</thead>'
print '<tbody>'

```

```

while @@fetch_status = 0
begin
    --print '<para>Hello !</para>'
    --print '<para>' + @l_index_column_name + ' : ' + t
    print '<row>'
    print '<entry>' + @l_index_column_name + '</entry>'
    print '<entry>' + type_name(@l_index_column_type)
    print '<entry>' + cast(@l_index_column_id as nvarchar)
    print '<entry>' + cast(@l_index_column_keyno as nvarchar)
    print '<entry>TODO</entry>'
    print '</row>'
    fetch next from index_columns_cursor
    into @l_index_column_name,
        @l_index_column_id,
        @l_index_column_type,
        @l_index_column_keyno

    end
    print '</tbody>'
    print '</tgroup>'
    print '</table>'
    close index_columns_cursor
    deallocate index_columns_cursor
    print '</para>'
    print '</section>'

    print '</section>'-- close section of current index

    fetch next from indexes_cursor into @l_index_status,
    @l_index_indid,
    @l_index_keycnt,
    @l_index_groupid,
    @l_index_rowcnt,
    @l_index_xmaxlen,
    @l_index_maxirow,
    @l_index_lockflags,
    @l_index_name

    end
    close indexes_cursor
    deallocate indexes_cursor
    print '</section>'
    end
    print '</section>'

else
    end
begin
    print '<warning>Table ' + @table_name + ' not found !!!</warning>'
end

end

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

```

sp_dbk_print_tables_book

Summary

Find below summarized details :

- Creation Date : Nov 13 2005 11:12AM

Parameters

No Input Parameters Required

Query Analyzer Call example

Find below how to call sp_dbk_print_tables_book procedure :

```
exec sp_dbk_print_tables_book
```

Source Code

Below find the TSQL statement used to create (or recreate)sp_dbk_print_tables_book procedure :

```
if exists(  
select * from dbo.sysobjects  
where id = object_id(N'[dbo].[sp_dbk_print_tables_book]')  
and OBJECTPROPERTY(id, N'IsProcedure') = 1)  
drop procedure [dbo].[sp_dbk_print_tables_book]
```

```
SET QUOTED_IDENTIFIER ON  
GO  
SET ANSI_NULLS ON  
GO
```

```
/*  
Copyright (C) 2005 Adrien Sales
```

```
This program is free software; you can redistribute it and/or  
modify it under the terms of the GNU General Public License  
as published by the Free Software Foundation.
```

```
This program is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License  
along with this program; if not, write to the Free Software  
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.
```

```
You can contact me by email at : prslump at msn dot com or at my job address :  
Adriens dot Sales at Experian-Scorex dot com, or still visit my own website :  
http://adrien.sales.free.fr/, by msn (mail and instant messaging at the following  
address : prslump@msn.com), or still by icq : 197582571
```

```
To print the GNU licence, you can call the sp_txt_print_gpl stored procedure  
on the standard Query Analyzer text output.  
*/
```

```
-- exec sp_dbk_print_table_book 'Products'

CREATE      procedure sp_dbk_print_tables_book
as
begin
declare @title nvarchar(300)
set @title = 'User Tables Report'
    -- print the docbookheader
    --exec sp_dbk_print_header
    print '<book>'
    exec sp_dbk_print_bookinfo @title
    -- print chapter
    exec sp_dbk_print_tables_chapter
    print '</book>'

end

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

sp_dbk_print_tables_chapter

Summary

Find below summarized details :

- Creation Date : Nov 13 2005 11:07AM

Parameters

No Input Parameters Required

Query Analyzer Call example

Find below how to call sp_dbk_print_tables_chapter procedure :

```
exec sp_dbk_print_tables_chapter
```

Source Code

Below find the TSQL statement used to create (or recreate)sp_dbk_print_tables_chapter procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[sp_dbk_print_tables_chapter]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[sp_dbk_print_tables_chapter]
```



```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO
```

```
/*
Copyright (C) 2005 Adrien Sales
```

```
This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation.
```

```
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.
```

```
You can contact me by email at : prslump at msn dot com or at my job address :
Adriens dot Sales at Experian-Scorex dot com, or still visit my own website :
http://adrien.sales.free.fr/, by msn (mail and instant messaging at the following
address : prslump@msn.com), or still by icq : 197582571
```

```
To print the GNU licence, you can call the sp_txt_print_gpl stored procedure
on the standard Query Analyzer text output.
*/
```

```
--
CREATE procedure sp_dbk_print_tables_chapter
as
begin
declare @table_name nvarchar(50)
declare @nb_tables int

select @nb_tables = count(*)from sysobjects where xtype = 'u'
if (@nb_tables > 0)
begin
    print '<chapter id = "table">'
    print '<title>Tables Report (' + cast(@nb_tables as nvarchar)+ ' tables)</
-- parse tables and build a section for eachof them
declare table_cursor cursor for
select name from sysobjects where xtype = 'u'
open table_cursor
fetch next from table_cursor into @table_name
while @@fetch_status = 0
begin
    exec sp_dbk_print_table_section @table_name
    fetch next from table_cursor into @table_name
end
close table_cursor
deallocate table_cursor
print '</chapter>'

end
end

GO
SET QUOTED_IDENTIFIER OFF
GO
```

```
SET ANSI_NULLS ON
GO
```

sp_dbk_print_types_chapter

Summary

Find below summarized details :

- Creation Date : Nov 27 2005 6:30PM

Parameters

No Input Parameters Required

Query Analyzer Call example

Find below how to call sp_dbk_print_types_chapter procedure :

```
exec sp_dbk_print_types_chapter
```

Source Code

Below find the TSQL statement used to create (or recreate)sp_dbk_print_types_chapter procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[sp_dbk_print_types_chapter]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[sp_dbk_print_types_chapter]
```

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO
```

```
/*
Copyright (C) 2005 Adrien Sales
```

```
This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation.
```

```
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.
```

You can contact me by email at : prslump at msn dot com or at my job address : Adriens dot Sales at Experian-Scorex dot com, or still visit my own website : <http://adrien.sales.free.fr/>, by msn (mail and instant messaging at the following address : prslump@msn.com), or still by icq : 197582571

To print the GNU licence, you can call the sp_txt_print_gpl stored procedure on the standard Query Analyzer text output.
*/

```

CREATE    procedure sp_dbk_print_types_chapter
as
begin

declare @xtype          int
declare @name           nvarchar(35)
declare @length         int
declare @xprec          int
declare @xscale         int
declare @allownulls int
declare @collation nvarchar(35)
declare @nb_types int

declare types_cursor cursor for
        select xtype,
               name,
               length,
               xprec,
               xscale,
               allownulls,
               isnull(collation, 'N/A')
        from systypes
        order by name asc

select @nb_types = count(*) from systypes

print '<chapter id = "types">'
print '<title>Types (' + cast(@nb_types as varchar) + ')</title>'
print '<section id = "types-summary">'
print '<title>List</title>'

open types_cursor
fetch next from types_cursor into
        @xtype,
        @name,
        @length,
        @xprec,
        @xscale,
        @allownulls,
        @collation
print '<table frame="all"><title>Types</title>'
print '<tgroup cols="7" align="left" colsep="1" rowsep="1">'
print '<colspec colname="c1"/>'
print '<colspec colname="c2"/>'
print '<colspec colname="c3"/>'
print '<colspec colname="c4"/>'
print '<colspec colname="c5"/>'
print '<colspec colname="c6"/>'
print '<colspec colname="c7"/>'

```

```

print '<spansec spanname="hspan" namest="c1" nameend="c2" align="center"/>'
print '<spansec spanname="hspan" namest="c2" nameend="c3" align="center"/>'
print '<spansec spanname="hspan" namest="c3" nameend="c4" align="center"/>'
print '<spansec spanname="hspan" namest="c4" nameend="c5" align="center"/>'
print '<spansec spanname="hspan" namest="c5" nameend="c6" align="center"/>'
print '<spansec spanname="hspan" namest="c6" nameend="c7" align="center"/>'
print '<thead>'
print '<row>'
print '<entry>Name</entry>'
print '<entry>xtype</entry>'
print '<entry>Length</entry>'
print '<entry>Precision</entry>'
print '<entry>Scale</entry>'
print '<entry>Allow Null</entry>'
print '<entry>Collation</entry>'
print '</row>'
print '</thead>'
print '<tbody>'

fetch next from types_cursor into
    @xtype,
    @name,
    @length,
    @xprec,
    @xscale,
    @allownulls,
    @collation
while @@fetch_status = 0
begin

    print '<row>'
    print '<entry>' + @name + '</entry>'
    print '<entry>' + cast(@xtype as varchar) + '</entry>'
    print '<entry>' + cast(@length as varchar) + '</entry>'
    print '<entry>' + cast(@xprec as varchar) + '</entry>'
    print '<entry>' + cast(@xscale as varchar) + '</entry>'
    if (@allownulls = 1)
    begin
        print '<entry>YES</entry>'
    end
    else
    begin
        print '<entry>NO</entry>'
    end
    print '<entry>' + @collation + '</entry>'
    print '</row>'
fetch next from types_cursor into
    @xtype,
    @name,
    @length,
    @xprec,
    @xscale,
    @allownulls,
    @collation

end

close types_cursor
deallocate types_cursor
print '</tbody>'
print '</tgroup>'
print '</table>'
print '</section>'
print '</chapter>'

```

```
end

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

sp_dbk_print_user_section

Summary

Find below summarized details :

- Creation Date : Nov 26 2005 6:00PM

Parameters

Find below input parameters :

1. @user_name : nvarchar (40)

Query Analyzer Call example

Find below how to call sp_dbk_print_user_section procedure :

```
/* Warning : do not forget to include declare section before to use the call example ! */
```

```
exec sp_dbk_print_user_section @user_name
```

Source Code

Below find the TSQL statement used to create (or recreate)sp_dbk_print_user_section procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[sp_dbk_print_user_section]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[sp_dbk_print_user_section]
```

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO
```

```
/*
Copyright (C) 2005 Adrien Sales
```

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

You can contact me by email at : prslump at msn dot com or at my job address : Adriens dot Sales at Experian-Scorex dot com, or still visit my own website : <http://adrien.sales.free.fr/>, by msn (mail and instant messaging at the following address : prslump@msn.com), or still by icq : 197582571

To print the GNU licence, you can call the sp_txt_print_gpl stored procedure on the standard Query Analyzer text output.

*/

```

CREATE procedure sp_dbk_print_user_section(@user_name nvarchar(20))
as
begin
declare @user_id int
declare @create_date datetime
declare @update_date datetime
declare @isnt_name int
declare @isnt_user int
declare @issql_user int
declare @hasdb_access int
if exists (select name from sysusers where name = @user_name and hasdbaccess = 1)
begin
    select @user_id = uid,
           @create_date = createdate,
           @update_date = updatedate,
           @hasdb_access = hasdbaccess,
           @isnt_name = isntname,
           @isnt_user = isntuser,
           @issql_user = issqluser
    from sysusers
    where name = @user_name

print '<section id = "users-' + cast(@user_id as varchar) + '">'
print '<title>' + @user_name+ '</title>'
print '<para>'
    -- print column headers
    print '<table frame="all"><title>' + @user_name + ' Detail
    print '<tgroup cols="2" align="left" colsep="1" rowsep="1"
    print '<colspec colname="c1"/>'
    print '<colspec colname="c2"/>'
    print '<spanspec spanname="hspan" namest="c1" nameend="c2"
    print '<thead>'
    print '<row>'
    print '<entry>Property</entry>'
    print '<entry>Value</entry>'
    print '</row>'
    print '</thead>'
    print '<tbody>'

    print '<row>'

```

```

print '<entry>Creation Date</entry>'
print '<entry>' + cast(@create_date as varchar) + '</entry>'
print '</row>'
print '<row>'
print '<entry>Update Date</entry>'
print '<entry>' + cast(@update_date as varchar)+ '</entry>'
print '</row>'
print '<row>'
print '<entry>Is NT Name</entry>'
print '<entry>' + cast(@isnt_name as varchar) + '</entry>'
print '</row>'
print '<row>'
print '<entry>NT User</entry>'
print '<entry>' + cast(@isnt_user as varchar) + '</entry>'
print '</row>'
print '<row>'
print '<entry>SQL Server User</entry>'
print '<entry>' + cast(@issql_user as varchar) + '</entry>'
print '</row>'
print '</tbody>'
print '</tgroup>'
print '</table>'

print '</para>'

print '</section>'
end

/*
select name,
       createdate,
       updatedate,
       hasdbaccess,
       isntname,
       isntuser,
       issqluser
from sysusers
where
--name not like 'public%'
-- and name not like 'db_%'
isntgroup = 0
and islogin = 1
order by uid asc

*/

end

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

```

sp_dbk_print_users_chapter

Summary

Find below summarized details :

- Creation Date : Nov 27 2005 6:09PM

Parameters

No Input Parameters Required

Query Analyzer Call example

Find below how to call sp_dbk_print_users_chapter procedure :

```
exec sp_dbk_print_users_chapter
```

Source Code

Below find the TSQL statement used to create (or recreate)sp_dbk_print_users_chapter procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[sp_dbk_print_users_chapter]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[sp_dbk_print_users_chapter]
```

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO
```

```
/*
Copyright (C) 2005 Adrien Sales
```

```
This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation.
```

```
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.
```

```
You can contact me by email at : prslump at msn dot com or at my job address :
Adriens dot Sales at Experian-Scorex dot com, or still visit my own website :
http://adrien.sales.free.fr/, by msn (mail and instant messaging at the following
address : prslump@msn.com), or still by icq : 197582571
```

```
To print the GNU licence, you can call the sp_txt_print_gpl stored procedure
on the standard Query Analyzer text output.
```

```
*/
CREATE procedure sp_dbk_print_users_chapter as

begin

declare @user_name nvarchar(25)
```



```
declare @nb_users int
declare users_cursor cursor for
    select name
    from sysusers
    where hasdbaccess = 1
if exists(select name from sysusers where hasdbaccess = 1)
begin
select @nb_users = count(*) from sysusers where hasdbaccess = 1
print '<chapter id = "users">'
print '<title>Users(' + cast(@nb_users as varchar) + ')</title>'
open users_cursor
fetch next from users_cursor into @user_name
while @@fetch_status = 0
begin
exec sp_dbk_print_user_section @user_name
fetch next from users_cursor into @user_name
end

close users_cursor
deallocate users_cursor
print '</chapter>'
end
end

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

sp_dbk_print_view_section

Summary

Find below summarized details :

- Creation Date : Nov 13 2005 3:17PM

Parameters

Find below input parameters :

1. @view_name : nvarchar (100)

Query Analyzer Call example

Find below how to call sp_dbk_print_view_section procedure :

```
/* Warning : do not forget to include declare section before to use the call example ! */
```

```
exec sp_dbk_print_view_section @view_name
```

Source Code

Below find the TSQL statement used to create (or recreate)sp_dbk_print_view_section procedure :

```

if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[sp_dbk_print_view_section]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[sp_dbk_print_view_section]

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

/*
Copyright (C) 2005  Adrien Sales

This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

You can contact me by email at : prslump at msn dot com or at my job address :
Adriens dot Sales at Experian-Scorex dot com, or still visit my own website :
http://adrien.sales.free.fr/, by msn (mail and instant messaging at the following
address : prslump@msn.com), or still by icq : 197582571

To print the GNU licence, you can call the sp_txt_print_gpl stored procedure
on the standard Query Analyzer text output.
*/

CREATE      procedure sp_dbk_print_view_section(@view_name nvarchar(50))
as
begin

declare @view_id int
declare @creation_date datetime

declare @l_column_name          nvarchar(50)
declare @l_column_xtype        int
declare @l_column_iscomputed int
declare @l_columns_length int
declare @l_column_xprec        int
declare @l_column_isnullable int
declare @l_column_collation nvarchar(50)

if exists(select id from sysobjects where xtype = 'V' and name = @view_name)
begin
        select @view_id = id,
               @creation_date = crdate
        from sysobjects
        where xtype = 'V'

```

```

        and name = @view_name
print '<section id = "view-' + cast(@view_id as nvarchar)+ '">'
print '<title>' + @view_name + ' </title>'
-- summary section
print '<section id = "view-' + cast(@view_id as nvarchar)+ '-summary">'
print '<title>Summary</title>'
print '<para>Find below summarized details of ' + @view_name + ' : '
print '<itemizedlist>'
print '<listitem>Creation Date : ' + cast(@creation_date as nvarchar)+ '</listitem>'
print '<listitem></listitem>'
print '<listitem></listitem>'
print '</itemizedlist>'
print '</para>'
print '</section>'-- end of summary section

-- column detailed section
print '<section id = "view-' + cast(@view_id as nvarchar)+ '-columns">'
print '<title>Columns</title>'

print '<table frame="all"><title>' + @view_name + ' Columns Details</title>'
print '<tgroup cols="7" align="left" colsep="1" rowsep="1">'
print '<colspec colname="c1"/>'
print '<colspec colname="c2"/>'
print '<colspec colname="c3"/>'
print '<colspec colname="c4"/>'
print '<colspec colname="c5"/>'
print '<colspec colname="c6"/>'
print '<colspec colname="c7"/>'
print '<spanspec spanname="hspan" namest="c1" nameend="c2" align="center"/>'
print '<spanspec spanname="hspan" namest="c2" nameend="c3" align="center"/>'
print '<spanspec spanname="hspan" namest="c3" nameend="c4" align="center"/>'
print '<spanspec spanname="hspan" namest="c4" nameend="c5" align="center"/>'
print '<spanspec spanname="hspan" namest="c5" nameend="c6" align="center"/>'
print '<spanspec spanname="hspan" namest="c6" nameend="c7" align="center"/>'
print '<spanspec spanname="hspan" namest="c7" nameend="c8" align="center"/>'
print '<thead>'
print '<row>'
print '<entry>Column Name</entry>'
print ' <entry>Type</entry>'
print ' <entry>Computed</entry>'
print ' <entry>Length</entry>'
print ' <entry>Prec</entry>'
print ' <entry>Nullable</entry>'
print ' <entry>Collation</entry>'
print '</row>'
print '</thead>'
print '<tbody>'
declare columns_cursor cursor for
select name,
       xtype,
       iscomputed,
       length,
       xprec,
       isnullable,
       collation
from syscolumns
where id = @view_id
order by colid asc
open columns_cursor
fetch next from columns_cursor into @l_column_name,
                                     @l_column_xtype,
                                     @l_column_iscomputed,
                                     @l_columns_length,
                                     @l_column_xprec,

```

```

        @l_column_isnullable,
        @l_column_collation
while @@fetch_status = 0
begin
print '<row>'
print '<entry>' + @l_column_name + '</entry>'
print '<entry>' + type_name(@l_column_xtype) + '</entry>'
if (@l_column_iscomputed = 0)
begin
    print '<entry>No</entry>'
end
else
begin
    print '<entry>Yes</entry>'
end
end

print '<entry>' + cast(@l_columns_length as nvarchar) + '</entry>'
print '<entry>' + cast(@l_column_xprec as nvarchar) + '</entry>'
if (@l_column_isnullable = 0)
begin
    print '<entry>No</entry>'
end
else
begin
    print '<entry>Yes</entry>'
end
end

print '<entry>' + @l_column_collation+ '</entry>'
print '</row>'

fetch next from columns_cursor into @l_column_name,
    @l_column_xtype,
    @l_column_iscomputed,
    @l_columns_length,
    @l_column_xprec,
    @l_column_isnullable,
    @l_column_collation

end

close columns_cursor
deallocate columns_cursor
print '</tbody>'
print '</tgroup>'
print '</table>'
print '</section>'-- end of columns section

-- source code section
print '<section id = "view-' + cast(@view_id as nvarchar)+ '-code">'
print '<title>Source Code</title>'
print '<para>Below find the TSQL statement used to create (or recreate)' +
declare @source nvarchar(255)
open source_cursor
print '<programlisting>'
print 'if exists (select * from dbo.sysobjects'
print 'where id = object_id(N''[' + user_name() + '].[' + @view_name + ']'
print 'and OBJECTPROPERTY(id, N''IsView'') = 1)'
print 'drop view [' + user_name() + '].[' + @view_name + ']'
print 'GO'
print ''
print 'SET QUOTED_IDENTIFIER ON '
print 'GO'
print 'SET ANSI_NULLS ON '

```

```
        print 'GO'
        print ''
        fetch next from source_cursor into @source
        while @@fetch_status = 0
        begin
            print @source
            fetch next from source_cursor into @source
        end
        close source_cursor
        deallocate source_cursor
        print 'GO'
        print 'SET QUOTED_IDENTIFIER OFF '
        print 'GO'
        print 'SET ANSI_NULLS ON'
        print 'GO'
        print '</programlisting>'
        print '</section>'
    print '</section>'
end

end

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

sp_dbk_print_views_book

Summary

Find below summarized details :

- Creation Date : Nov 14 2005 10:31AM

Parameters

No Input Parameters Required

Query Analyzer Call example

Find below how to call sp_dbk_print_views_book procedure :

```
exec sp_dbk_print_views_book
```

Source Code

Below find the TSQL statement used to create (or recreate)sp_dbk_print_views_book procedure :

```
if exists(
select * from dbo.sysobjects
```

```
where id = object_id(N'[dbo].[sp_dbk_print_views_book]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[sp_dbk_print_views_book]
```

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO
```

```
/*
Copyright (C) 2005 Adrien Sales
```

```
This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation.
```

```
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.
```

```
You can contact me by email at : prslump at msn dot com or at my job address :
Adriens dot Sales at Experian-Scorex dot com, or still visit my own website :
http://adrien.sales.free.fr/, by msn (mail and instant messaging at the following
address : prslump@msn.com), or still by icq : 197582571
```

```
To print the GNU licence, you can call the sp_txt_print_gpl stored procedure
on the standard Query Analyzer text output.
*/
```

```
CREATE procedure sp_dbk_print_views_book
as
begin

declare @title nvarchar(50)

set @title = 'User Views Report'
-- print the docbookheader
--exec sp_dbk_print_header
print '<book>'
exec sp_dbk_print_bookinfo @title
-- print chapter
exec sp_dbk_print_views_chapter
print '</book>'

end

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

sp_dbk_print_views_chapter

Summary

Find below summarized details :

- Creation Date : Nov 14 2005 10:31AM

Parameters

No Input Parameters Required

Query Analyzer Call example

Find below how to call sp_dbk_print_views_chapter procedure :

```
exec sp_dbk_print_views_chapter
```

Source Code

Below find the TSQL statement used to create (or recreate)sp_dbk_print_views_chapter procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[sp_dbk_print_views_chapter]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[sp_dbk_print_views_chapter]
```

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO
```

```
/*
Copyright (C) 2005 Adrien Sales
```

```
This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation.
```

```
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.
```

```
You can contact me by email at : prslump at msn dot com or at my job address :
Adriens dot Sales at Experian-Scorex dot com, or still visit my own website :
http://adrien.sales.free.fr/, by msn (mail and instant messaging at the following
address : prslump@msn.com), or still by icq : 197582571
```

```
To print the GNU licence, you can call the sp_txt_print_gpl stored procedure
```

```
on the standard Query Analyzer text output.  
*/
```

```
CREATE          procedure sp_dbk_print_views_chapter  
as  
begin  
  declare @view_name          nvarchar(50)  
  declare @nb_views int  
  if exists (select id from sysobjects where xtype = 'V' and name not like 'sys%')  
  begin  
    select @nb_views = count(*) from sysobjects where xtype = 'V' and name not  
    print '<chapter id = "view">'  
    print '<title>Views (' + cast(@nb_views as nvarchar)+ ' Views)</title>'  
    declare views_cursor cursor for  
    select name from sysobjects  
    where xtype = 'V'  
    and name not like 'sys%'  
    order by name asc  
    open views_cursor  
    fetch next from views_cursor into @view_name  
    while @@fetch_status = 0  
    begin  
      exec sp_dbk_print_view_section @view_name  
      fetch next from views_cursor into @view_name  
    end  
    close views_cursor  
    deallocate views_cursor  
    print '</chapter>'  
  end  
end  
--sp_dbk_print_view_section 'vw_toto'  
end
```

```
GO  
SET QUOTED_IDENTIFIER OFF  
GO  
SET ANSI_NULLS ON  
GO
```

sp_print_dbk_project_details_section

Summary

Find below summarized details :

- Creation Date : Nov 27 2005 7:03PM

Parameters

No Input Parameters Required

Query Analyzer Call example

Find below how to call sp_print_dbk_project_details_section procedure :

```
exec sp_print_dbk_project_details_section
```

Source Code

Below find the TSQL statement used to create (or recreate)sp_print_dbk_project_details_section procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[sp_print_dbk_project_details_section]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[sp_print_dbk_project_details_section]
```

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO
```

```
/*
Copyright (C) 2005 Adrien Sales
```

```
This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation.
```

```
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.
```

```
You can contact me by email at : prslump at msn dot com or at my job address :
Adriens dot Sales at Experian-Scorex dot com, or still visit my own website :
http://adrien.sales.free.fr/, by msn (mail and instant messaging at the following
address : prslump@msn.com), or still by icq : 197582571
```

```
To print the GNU licence, you can call the sp_txt_print_gpl stored procedure
on the standard Query Analyzer text output.
```

```
*/
```

```
CREATE procedure sp_print_dbk_project_details_section
as
begin
print '<section id = "project-details">'
print '<title>Details</title>'
print '<para><ulink url = "http://sqlserverbook.sourceforge.net/">SQLServer Report
is an open source project hosted by <ulink url = "http://sourceforge.net/">Sourcef
print '</section>'
```

```
end
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

sp_print_dbk_project_info_chapter

Summary

Find below summarized details :

- Creation Date : Nov 27 2005 6:59PM

Parameters

No Input Parameters Required

Query Analyzer Call example

Find below how to call sp_print_dbk_project_info_chapter procedure :

```
exec sp_print_dbk_project_info_chapter
```

Source Code

Below find the TSQL statement used to create (or recreate)sp_print_dbk_project_info_chapter procedure :

```
if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[sp_print_dbk_project_info_chapter]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[sp_print_dbk_project_info_chapter]
```

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO
```

```
/*
Copyright (C) 2005 Adrien Sales
```

```
This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation.
```

```
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
```

MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

You can contact me by email at : prslump at msn dot com or at my job address : Adriens dot Sales at Experian-Scorex dot com, or still visit my own website : <http://adrien.sales.free.fr/>, by msn (mail and instant messaging at the following address : prslump@msn.com), or still by icq : 197582571

To print the GNU licence, you can call the sp_txt_print_gpl stored procedure on the standard Query Analyzer text output.
*/

```
CREATE procedure sp_print_dbk_project_info_chapter
as
begin
print '<chapter id = "project">'
print '<title>SQLServer Reports Project Info</title>'
exec sp_print_dbk_project_details_section
print '</chapter>'
end
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

sp_txt_print_gpl

Summary

Find below summarized details :

- Creation Date : Nov 14 2005 10:21AM

Parameters

No Input Parameters Required

Query Analyzer Call example

Find below how to call sp_txt_print_gpl procedure :

```
exec sp_txt_print_gpl
```

Source Code

Below find the TSQL statement used to create (or recreate)sp_txt_print_gpl procedure :

```

if exists(
select * from dbo.sysobjects
where id = object_id(N'[dbo].[sp_txt_print_gpl]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[sp_txt_print_gpl]

```

```

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

```

```

/*
Copyright (C) 2005 Adrien Sales

```

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

You can contact me by email at : prslump at msn dot com or at my job address : Adriens dot Sales at Experian-Scorex dot com, or still visit my own website : <http://adrien.sales.free.fr/>, by msn (mail and instant messaging at the following address : prslump@msn.com), or still by icq : 197582571

To print the GNU licence, you can call the sp_txt_print_gpl stored procedure on the standard Query Analyzer text output.
*/

```

CREATE procedure sp_txt_print_gpl
as
begin
print '          GNU GENERAL PUBLIC LICENSE'
print '          Version 2, June 1991'
print ''
print ' Copyright (C) 1989, 1991 Free Software Foundation, Inc.'
print '          51 Franklin St, Fifth Floor, Boston, MA 02110-1301'
print ' Everyone is permitted to copy and distribute verbatim copies'
print ' of this license document, but changing it is not allowed.'
print ''
print '          Preamble'
print ''
print ' The licenses for most software are designed to take away your'
print ' freedom to share and change it. By contrast, the GNU General Public'
print ' License is intended to guarantee your freedom to share and change free'
print ' software--to make sure the software is free for all its users. This'
print ' General Public License applies to most of the Free Software'
print ' Foundation s software and to any other program whose authors commit to'
print ' using it. (Some other Free Software Foundation software is covered by'
print ' the GNU Library General Public License instead.) You can apply it to'
print ' your programs, too.'
print ''
print ' When we speak of free software, we are referring to freedom, not'
print ' price. Our General Public Licenses are designed to make sure that you'
print ' have the freedom to distribute copies of free software (and charge for'

```

```

print 'this service if you wish), that you receive source code or can get it'
print 'if you want it, that you can change the software or use pieces of it'
print 'in new free programs; and that you know you can do these things.'
print ''
print ' To protect your rights, we need to make restrictions that forbid'
print 'anyone to deny you these rights or to ask you to surrender the rights.'
print 'These restrictions translate to certain responsibilities for you if you'
print 'distribute copies of the software, or if you modify it.'
print ''
print ' For example, if you distribute copies of such a program, whether'
print 'gratis or for a fee, you must give the recipients all the rights that'
print 'you have. You must make sure that they, too, receive or can get the'
print 'source code. And you must show them these terms so they know their'
print 'rights.'
print ''
print ' We protect your rights with two steps: (1) copyright the software, and'
print '(2) offer you this license which gives you legal permission to copy,'
print 'distribute and/or modify the software.'
print ''
print ' Also, for each author s protection and ours, we want to make certain'
print 'that everyone understands that there is no warranty for this free'
print 'software. If the software is modified by someone else and passed on, we'
print 'want its recipients to know that what they have is not the original, so'
print 'that any problems introduced by others will not reflect on the original'
print 'authors reputations.'
print ''
print ' Finally, any free program is threatened constantly by software'
print 'patents. We wish to avoid the danger that redistributors of a free'
print 'program will individually obtain patent licenses, in effect making the'
print 'program proprietary. To prevent this, we have made it clear that any'
print 'patent must be licensed for everyone s free use or not licensed at all.'
print ''
print ' The precise terms and conditions for copying, distribution and'
print 'modification follow.'
print ''
print '          GNU GENERAL PUBLIC LICENSE'
print '    TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION'
print ''
print ' 0. This License applies to any program or other work which contains'
print 'a notice placed by the copyright holder saying it may be distributed'
print 'under the terms of this General Public License. The "Program", below,'
print 'refers to any such program or work, and a "work based on the Program"'
print 'means either the Program or any derivative work under copyright law:'
print 'that is to say, a work containing the Program or a portion of it,'
print 'either verbatim or with modifications and/or translated into another'
print 'language. (Hereinafter, translation is included without limitation in'
print 'the term "modification".) Each licensee is addressed as "you".'
print ''
print 'Activities other than copying, distribution and modification are not'
print 'covered by this License; they are outside its scope. The act of'
print 'running the Program is not restricted, and the output from the Program'
print 'is covered only if its contents constitute a work based on the'
print 'Program (independent of having been made by running the Program).'
print 'Whether that is true depends on what the Program does.'
print ''
print ' 1. You may copy and distribute verbatim copies of the Program s'
print 'source code as you receive it, in any medium, provided that you'
print 'conspicuously and appropriately publish on each copy an appropriate'
print 'copyright notice and disclaimer of warranty; keep intact all the'
print 'notices that refer to this License and to the absence of any warranty;'
print 'and give any other recipients of the Program a copy of this License'
print 'along with the Program.'
print ''
print 'You may charge a fee for the physical act of transferring a copy, and'

```

```

print 'you may at your option offer warranty protection in exchange for a fee.'
print ''
print '  2. You may modify your copy or copies of the Program or any portion'
print 'of it, thus forming a work based on the Program, and copy and'
print 'distribute such modifications or work under the terms of Section 1'
print 'above, provided that you also meet all of these conditions:'
print ''
print '    a) You must cause the modified files to carry prominent notices'
print '    stating that you changed the files and the date of any change.'
print ''
print '    b) You must cause any work that you distribute or publish, that in'
print '    whole or in part contains or is derived from the Program or any'
print '    part thereof, to be licensed as a whole at no charge to all third'
print '    parties under the terms of this License.'
print ''
print '    c) If the modified program normally reads commands interactively'
print '    when run, you must cause it, when started running for such'
print '    interactive use in the most ordinary way, to print or display an'
print '    announcement including an appropriate copyright notice and a'
print '    notice that there is no warranty (or else, saying that you provide'
print '    a warranty) and that users may redistribute the program under'
print '    these conditions, and telling the user how to view a copy of this'
print '    License. (Exception: if the Program itself is interactive but'
print '    does not normally print such an announcement, your work based on'
print '    the Program is not required to print an announcement.)'
print ''
print 'These requirements apply to the modified work as a whole.  If'
print 'identifiable sections of that work are not derived from the Program,'
print 'and can be reasonably considered independent and separate works in'
print 'themselves, then this License, and its terms, do not apply to those'
print 'sections when you distribute them as separate works.  But when you'
print 'distribute the same sections as part of a whole which is a work based'
print 'on the Program, the distribution of the whole must be on the terms of'
print 'this License, whose permissions for other licensees extend to the'
print 'entire whole, and thus to each and every part regardless of who wrote it.'
print ''
print 'Thus, it is not the intent of this section to claim rights or contest'
print 'your rights to work written entirely by you; rather, the intent is to'
print 'exercise the right to control the distribution of derivative or'
print 'collective works based on the Program.'
print ''
print 'In addition, mere aggregation of another work not based on the Program'
print 'with the Program (or with a work based on the Program) on a volume of'
print 'a storage or distribution medium does not bring the other work under'
print 'the scope of this License.'
print ''
print '  3. You may copy and distribute the Program (or a work based on it,'
print 'under Section 2) in object code or executable form under the terms of'
print 'Sections 1 and 2 above provided that you also do one of the following:'
print ''
print '    a) Accompany it with the complete corresponding machine-readable'
print '    source code, which must be distributed under the terms of Sections'
print '    1 and 2 above on a medium customarily used for software interchange; or'
print ''
print '    b) Accompany it with a written offer, valid for at least three'
print '    years, to give any third party, for a charge no more than your'
print '    cost of physically performing source distribution, a complete'
print '    machine-readable copy of the corresponding source code, to be'
print '    distributed under the terms of Sections 1 and 2 above on a medium'
print '    customarily used for software interchange; or,'
print ''
print '    c) Accompany it with the information you received as to the offer'
print '    to distribute corresponding source code. (This alternative is'
print '    allowed only for noncommercial distribution and only if you'

```

```

print '    received the program in object code or executable form with such'
print '    an offer, in accord with Subsection b above.)'
print ''
print 'The source code for a work means the preferred form of the work for'
print 'making modifications to it. For an executable work, complete source'
print 'code means all the source code for all modules it contains, plus any'
print 'associated interface definition files, plus the scripts used to'
print 'control compilation and installation of the executable. However, as a'
print 'special exception, the source code distributed need not include'
print 'anything that is normally distributed (in either source or binary'
print 'form) with the major components (compiler, kernel, and so on) of the'
print 'operating system on which the executable runs, unless that component'
print 'itself accompanies the executable.'
print ''
print 'If distribution of executable or object code is made by offering'
print 'access to copy from a designated place, then offering equivalent'
print 'access to copy the source code from the same place counts as'
print 'distribution of the source code, even though third parties are not'
print 'compelled to copy the source along with the object code.'
print ''
print ' 4. You may not copy, modify, sublicense, or distribute the Program'
print 'except as expressly provided under this License. Any attempt'
print 'otherwise to copy, modify, sublicense or distribute the Program is'
print 'void, and will automatically terminate your rights under this License.'
print 'However, parties who have received copies, or rights, from you under'
print 'this License will not have their licenses terminated so long as such'
print 'parties remain in full compliance.'
print ''
print ' 5. You are not required to accept this License, since you have not'
print 'signed it. However, nothing else grants you permission to modify or'
print 'distribute the Program or its derivative works. These actions are'
print 'prohibited by law if you do not accept this License. Therefore, by'
print 'modifying or distributing the Program (or any work based on the'
print 'Program), you indicate your acceptance of this License to do so, and'
print 'all its terms and conditions for copying, distributing or modifying'
print 'the Program or works based on it.'
print ''
print ' 6. Each time you redistribute the Program (or any work based on the'
print 'Program), the recipient automatically receives a license from the'
print 'original licensor to copy, distribute or modify the Program subject to'
print 'these terms and conditions. You may not impose any further'
print 'restrictions on the recipients exercise of the rights granted herein.'
print 'You are not responsible for enforcing compliance by third parties to'
print 'this License.'
print ''
print ' 7. If, as a consequence of a court judgment or allegation of patent'
print 'infringement or for any other reason (not limited to patent issues),'
print 'conditions are imposed on you (whether by court order, agreement or'
print 'otherwise) that contradict the conditions of this License, they do not'
print 'excuse you from the conditions of this License. If you cannot'
print 'distribute so as to satisfy simultaneously your obligations under this'
print 'License and any other pertinent obligations, then as a consequence you'
print 'may not distribute the Program at all. For example, if a patent'
print 'license would not permit royalty-free redistribution of the Program by'
print 'all those who receive copies directly or indirectly through you, then'
print 'the only way you could satisfy both it and this License would be to'
print 'refrain entirely from distribution of the Program.'
print ''
print 'If any portion of this section is held invalid or unenforceable under'
print 'any particular circumstance, the balance of the section is intended to'
print 'apply and the section as a whole is intended to apply in other'
print 'circumstances.'
print ''
print 'It is not the purpose of this section to induce you to infringe any'

```

```

print 'patents or other property right claims or to contest validity of any'
print 'such claims; this section has the sole purpose of protecting the'
print 'integrity of the free software distribution system, which is'
print 'implemented by public license practices. Many people have made'
print 'generous contributions to the wide range of software distributed'
print 'through that system in reliance on consistent application of that'
print 'system; it is up to the author/donor to decide if he or she is willing'
print 'to distribute software through any other system and a licensee cannot'
print 'impose that choice.'
print ''
print 'This section is intended to make thoroughly clear what is believed to'
print 'be a consequence of the rest of this License.'
print ''
print ' 8. If the distribution and/or use of the Program is restricted in'
print 'certain countries either by patents or by copyrighted interfaces, the'
print 'original copyright holder who places the Program under this License'
print 'may add an explicit geographical distribution limitation excluding'
print 'those countries, so that distribution is permitted only in or among'
print 'countries not thus excluded. In such case, this License incorporates'
print 'the limitation as if written in the body of this License.'
print ''
print ' 9. The Free Software Foundation may publish revised and/or new versions'
print 'of the General Public License from time to time. Such new versions will'
print 'be similar in spirit to the present version, but may differ in detail to'
print 'address new problems or concerns.'
print ''
print 'Each version is given a distinguishing version number. If the Program'
print 'specifies a version number of this License which applies to it and "any'
print 'later version", you have the option of following the terms and conditions'
print 'either of that version or of any later version published by the Free'
print 'Software Foundation. If the Program does not specify a version number of'
print 'this License, you may choose any version ever published by the Free Softwar'
print 'Foundation.'
print ''
print ' 10. If you wish to incorporate parts of the Program into other free'
print 'programs whose distribution conditions are different, write to the author'
print 'to ask for permission. For software which is copyrighted by the Free'
print 'Software Foundation, write to the Free Software Foundation; we sometimes'
print 'make exceptions for this. Our decision will be guided by the two goals'
print 'of preserving the free status of all derivatives of our free software and'
print 'of promoting the sharing and reuse of software generally.'
print ''
print '          NO WARRANTY'
print ''
print ' 11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY'
print 'FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN'
print 'OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES'
print 'PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED'
print 'OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF'
print 'MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS'
print 'TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE'
print 'PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,'
print 'REPAIR OR CORRECTION.'
print ''
print ' 12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING'
print 'WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR'
print 'REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES,'
print 'INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING'
print 'OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED'
print 'TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY'
print 'YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER'
print 'PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE'
print 'POSSIBILITY OF SUCH DAMAGES.'
print ''

```



```
print '          END OF TERMS AND CONDITIONS'  
print ''  
end  
  
GO  
SET QUOTED_IDENTIFIER OFF  
GO  
SET ANSI_NULLS ON  
GO
```

toto

Summary

Find below summarized details :

- Creation Date : Nov 13 2005 1:19PM

Parameters

Find below input parameters :

1. @toto : int (4)
2. @titi : int (4)

Query Analyzer Call example

Find below how to call toto procedure :

```
/* Warning : do not forget to include declare section before to use the call example ! */
```

```
exec toto @toto, @titi
```

Source Code

Below find the TSQL statement used to create (or recreate)toto procedure :

```
if exists(  
select * from dbo.sysobjects  
where id = object_id(N'[dbo].[toto]')  
and OBJECTPROPERTY(id, N'IsProcedure') = 1)  
drop procedure [dbo].[toto]  
  
SET QUOTED_IDENTIFIER ON  
GO  
SET ANSI_NULLS ON  
GO
```

```
create procedure toto(@toto int, @titi int)
as
begin
    print 'hello'
end
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

Chapter 6. Functions (4 functs.)

test_function

Summary

Find below summarized details :

- Creation Date : Nov 13 2005 11:27AM

Input Parameters (1)

Find below input parameters details :

- @in : int(4)

Output Parameter

Find below details about the output parameter (required for any function) :

- Type : int
- Length : 4
- Precision : 10

Source Code

Below find the TSQL statement used to create (or recreate)test_function function :

```
if exists (select * from dbo.sysobjects
where id = object_id(N'[dbo].[test_function]')
and xtype in (N'FN', N'IF', N'TF'))
drop function [dbo].[test_function]
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS OFF
GO
create function test_function(@in int)
returns int
as
begin
    return 1
end
GO
SET QUOTED_IDENTIFIER OFF
```

```
GO
SET ANSI_NULLS ON
GO
```

fx_test

Summary

Find below summarized details :

- Creation Date : Nov 13 2005 1:52PM

Input Parameters (1)

Find below input parameters details :

- @in : int(4)

Output Parameter

Find below details about the output parameter (required for any function) :

- Type : int
- Length : 4
- Precision : 10

Source Code

Below find the TSQL statement used to create (or recreate)fx_test function :

```
if exists (select * from dbo.sysobjects
where id = object_id(N'[dbo].[fx_test]')
and xtype in (N'FN', N'IF', N'TF'))
drop function [dbo].[fx_test]
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS OFF
GO
create function fx_test(@in int)
returns int
as
begin

return 0
```

```

end
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

```

fx_test2

Summary

Find below summarized details :

- Creation Date : Nov 13 2005 2:45PM

Input Parameters (2)

Find below input parameters details :

- @toto : int(4)
- @titi : int(4)

Output Parameter

Find below details about the output parameter (required for any function) :

- Type : int
- Length : 4
- Precision : 10

Source Code

Below find the TSQL statement used to create (or recreate)fx_test2 function :

```

if exists (select * from dbo.sysobjects
where id = object_id(N'[dbo].[fx_test2]')
and xtype in (N'FN', N'IF', N'TF'))
drop function [dbo].[fx_test2]
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS OFF
GO
create function fx_test2(@toto int, @titi int)
returns int

```

```

as
begin
return 0
end

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

```

fx_dbk_get_xmlentities

Summary

Find below summarized details :

- Creation Date : Dec 8 2005 11:32AM

Input Parameters (1)

Find below input parameters details :

- @text : nvarchar(600)

Output Parameter

Find below details about the output parameter (required for any function) :

- Type : nvarchar
- Length : 2400
- Precision : 0
- Collation : SQL_Latin1_General_CP1_CI_AS

Source Code

Below find the TSQL statement used to create (or recreate)fx_dbk_get_xmlentities function :

```

if exists (select * from dbo.sysobjects
where id = object_id(N'[dbo].[fx_dbk_get_xmlentities]')
and xtype in (N'FN', N'IF', N'TF'))
drop function [dbo].[fx_dbk_get_xmlentities]
GO

SET QUOTED_IDENTIFIER OFF
GO

```

```
SET ANSI_NULLS OFF
GO
```

```
/*
Copyright (C) 2005 Adrien Sales
```

```
This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation.
```

```
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.
```

```
You can contact me by email at : prslump at msn dot com or at my job address :
Adriens dot Sales at Experian-Scorex dot com, or still visit my own website :
http://adrien.sales.free.fr/, by msn (mail and instant messaging at the following
address : prslump@msn.com), or still by icq : 197582571
```

```
To print the GNU licence, you can call the sp_txt_print_gpl stored procedure
on the standard Query Analyzer text output.
*/
```

```
CREATE      function fx_dbk_get_xmlentities(@text nvarchar(300))
returns nvarchar(1200)
as
```

```
begin
declare @out nvarchar(1200)
set @out = @text

-- replace < symbol
set @out = replace(@out,'<','<')

-- replace > symbol
set @out = replace(@out,'>','>')
```

```
return @out
end
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

Chapter 7. Types (26)

List

Table 7.1. Types

Name	xtype	Length	Precision	Scale	Allow Null	Collation
binary	173	8000	0	0	YES	N/A
bit	104	1	1	0	YES	N/A
char	175	8000	0	0	YES	SQL_Latin1_General_CP1_CI_AS
datetime	61	8	23	3	YES	N/A
decimal	106	17	38	38	YES	N/A
float	62	8	53	0	YES	N/A
image	34	16	0	0	YES	N/A
int	56	4	10	0	YES	N/A
money	60	8	19	4	YES	N/A
nchar	239	8000	0	0	YES	SQL_Latin1_General_CP1_CI_AS
ntext	99	16	0	0	YES	SQL_Latin1_General_CP1_CI_AS
numeric	108	17	38	38	YES	N/A
nvarchar	231	8000	0	0	YES	SQL_Latin1_General_CP1_CI_AS
real	59	4	24	0	YES	N/A
smalldatetime	58	4	16	0	YES	N/A
smallint	52	2	5	0	YES	N/A
smallmoney	122	4	10	4	YES	N/A
sql_variant	98	8016	0	0	YES	N/A
sysname	231	256	0	0	NO	SQL_Latin1_General_CP1_CI_AS
text	35	16	0	0	YES	SQL_Latin1_General_CP1_CI_AS
timestamp	189	8	0	0	NO	N/A
tinyint	48	1	3	0	YES	N/A
uniqueidentifier	36	16	0	0	YES	N/A
varbinary	165	8000	0	0	YES	N/A

Types (26)

Name	xtype	Length	Precision	Scale	Allow Null	Collation
varchar	167	8000	0	0	YES	SQL_Latin1_G ener- al_CP1_CI_AS

Chapter 8. Storage (1 filegroups)

PRIMARY filegroup

Details about PRIMARY filegroup

Summary

Below, find summarized details of PRIMARY filegroup :

- Total size : TODO
-

Data Files

dbreport_Data File

Find below summarized details of this file :

- File name : C:\Program Files\Microsoft SQL Server\MSSQL\data\dbreport_Data.MDF
- Max size : -1
- Growth : 10
- Status : 1081346
- Perf : 0

Chapter 9. Users(3)

AdrienS

Table 9.1. AdrienS Details

Property	Value
Creation Date	Jul 6 2005 5:14PM
Update Date	Jul 6 2005 5:14PM
Is NT Name	1
NT User	1
SQL Server User	0

dbo

Table 9.2. dbo Details

Property	Value
Creation Date	Aug 6 2000 1:27AM
Update Date	Aug 6 2000 1:27AM
Is NT Name	1
NT User	1
SQL Server User	0

toto

Table 9.3. toto Details

Property	Value
Creation Date	Dec 8 2005 2:22PM
Update Date	Dec 8 2005 2:22PM
Is NT Name	0
NT User	0
SQL Server User	1

Chapter 10. Management scripts

A set of scripts to make statistics, diagnose, admin tasks, ...

Tables Statistics Scripts