



APIIS Developer Documentation

written by

Eildert Groeneveld
Zhivko Ducheve
Ralf Fischer
Marek Imialek
Helmut Lichtenberg
Detlef Schulze

Institute of Farm Animal Genetics
friedrich Loeffler Institute
Mariensee, Germany
October 2008

Contents

1	Introduction	33
2	The APIIS Core	35
2.1	Basic Initialization	35
2.1.1	initialize	36
2.1.2	Apiis::Init	36
2.2	The Model File Interface	37
2.3	The Database Interface	39
2.4	The Database Record Object	40
2.4.1	The Column Object	40
3	The Meta-layer	43
3.1	Data-flow schema	43
3.2	PseudoSQL language definition	43
3.2.1	INSERT	43
3.2.2	UPDATE	45
3.2.3	DELETE	45
3.2.4	SELECT	45
3.3	PseudoSQL flow	46
3.4	Standard SQL flow	46
3.5	Used record object methods	48
4	The Access Control	51
4.1	Introduction	51
4.2	Requirements for the access control system	51
4.2.1	General requirements	51
4.2.2	Software requirements	52
4.2.3	Database requirements	52
4.3	The basic foundations - setting APIIS software	52
4.3.1	System architecture	52
4.3.2	APIIS core	53
4.3.3	APIIS projects	53
4.4	Defining users	54
4.4.1	Registering user on the operating system level	55
4.4.2	Registering user on the APIIS system level	55
4.5	Granting access rights to the user	56
4.6	Access rights for the system tasks	58
4.6.1	Definitions of the access rights	58

4.6.2	Checking of the access rights - logging to the system . . .	59
4.7	Access rights for the database and the content of the database .	60
4.7.1	Method for the insert, update and delete statements . . .	60
4.7.1.1	Definition of the access rights	61
4.7.1.2	Checking of the access rights	63
4.7.2	Method for the public select statements	66
4.7.2.1	Definition of the access rights:	66
4.7.2.2	Creating views:	66
4.8	Grouping access rights	68
4.8.1	Grouping roles	70
4.8.2	Grouping groups	70
4.9	Specifying constraints for the grouping	71
4.10	Further developing	73
4.10.1	Checking the login time and the current status of the users	73
4.11	Remarks	73
4.12	Bibliography	73
4.13	ERD diagrams	73
5	Synchronization of Database Content	79
5.1	Database structure	79
5.1.1	Additional fields	79
5.1.2	Additional tables	79
6	Internationalization in APIIS	81
6.1	Localization of Interfaces	81
6.2	Multilanguage Handling of Database Content	81
6.2.1	Implementation	81
6.2.1.1	Database Structure	81
6.2.1.2	Encoding	82
6.2.2	Meta level	83
6.2.3	Access rights	83
6.2.4	Synchronization	84
6.2.5	Outputs	84
6.3	International Character Sets	84
6.3.1	UNICODE	84
6.3.2	Support for UNICODE	84
6.3.3	Locales and fonts	85
6.4	The Core	85
6.4.1	Apiis.pm	85
6.4.2	Apiis::Init – Basic initialisation object for the complete APIIS structure	86
6.4.3	Apiis::Init::Config mainly ready apiisrc config files	95
6.4.4	SYNOPSIS	97
6.4.5	DESCRIPTION	97
6.4.6	METHODS	98
6.4.7	Apiis::Model – methods to access the model file data via the \$apiis structure	98
6.4.8	Apiis::Errors – Provide error objects for generic error han- dling in APIIS	101

6.4.9	Apiis::Misc – Provides some usefull subroutines, mainly for compatibility reasons	104
6.4.10	Apiis::CheckFile – Find configuration files	107
6.5	Authentication and Authorisation	107
6.5.1	DESCRIPTION	107
6.5.2	SUBROUTINES	107
6.5.3	AUTHORS	108
6.5.4	Apiis::Auth::Role.pm	108
6.5.5	Apiis::Auth::AppAuth – object for provading data about user access rights for the applications	109
6.5.6	Apiis::Auth::AccessControl – used by the runall.pl and access_control.pl scripts to define user access rights	110
6.6	The Database	113
6.6.1	Apiis::DataBase::Init – Basic database initialisation	113
6.6.2	DBCcreation.pm	115
6.6.3	Apiis::DataBase::User – collecting and providing user data	115
6.6.4	The Database Record Object	117
6.6.4.1	Apiis::DataBase::Record – package for DataBase Record objects	117
6.6.4.2	Apiis::DataBase::Record::Column – package for DataBase Record columns	119
6.6.4.3	Fetch	119
6.6.5	Modify Rules	121
6.6.5.1	CommaToDot	121
6.6.5.2	ConvertBool	121
6.6.5.3	DotToColon	121
6.6.5.4	LowerCase	122
6.6.5.5	SetNow	122
6.6.5.6	SetUser	122
6.6.5.7	UpperCase	122
6.6.6	Check Rules	122
6.6.6.1	DateDiff	122
6.6.6.2	ForeignKey	123
6.6.6.3	IsAFloat	124
6.6.6.4	IsANumber	124
6.6.6.5	IsEqual	125
6.6.6.6	AUTHORS	126
6.6.6.7	List	126
6.6.6.8	NoCheck	127
6.6.6.9	NoNumber	127
6.6.6.10	NotNull	128
6.6.6.11	Range	128
6.6.6.12	ReservedStrings	128
6.6.6.13	Unique	129
6.6.7	Triggers	129
6.6.7.1	SetGuid	129
6.6.7.2	SetNode	130
6.6.7.3	SetVersion	130
6.6.8	Using SQL	130
6.6.8.1	DataStream	130

6.6.8.2	Apiis::DataBase::SQL::DirectSQL Direct access to SQL	132
6.6.8.3	PseudoStatement	133
6.6.8.4	Statement	134
6.6.8.5	Apiis::DataBase::MakeSQL Module to create SQL-statements from the model file	135
6.6.9	The Synchronization	135
6.6.9.1	SYNOPSIS	135
6.6.9.2	PUBLIC METHODS	135
6.6.9.3	AUTHORS	136
6.7	The Forms	136
6.7.1	Apiis::Form::Init – base package for Form objects of all types	136
6.7.2	The Forms old	140
6.8	The Misc	141
6.8.1	usage	141
6.8.2	description	141
6.8.3	configuration	141
6.8.4	used in	141
6.8.5	usage	142
6.8.6	return	142
6.8.7	description	142
6.8.8	xfig_lib.pm	142
6.9	The Binaries	143
6.9.1	check_integrity	143
6.9.2	cvs2cl.pl	145
6.9.3	file2inspool.pl	152
6.9.4	Form	152
6.9.5	FormDesigner	153
6.9.6	mkform	158
6.9.7	create a formatted form file from each given loadobject by parsing the loadobject file about variable @LO_keys	159
6.9.8	mkLOform	159
6.9.9	mksql – create SQL commands from the model file	160
6.9.10	model2xfig	160
6.9.11	show_rules	161
6.9.12	WebForm.pl	162
6.9.13	163
7	Undocumented Subroutines	165
7.1	APIIS_HOME/lib/todo.js	165
7.2	APIIS_HOME/lib/menu_tpl1.js	165
7.3	APIIS_HOME/lib/apiis2css.js	165
7.4	APIIS_HOME/lib/form.js.org	165
7.5	APIIS_HOME/lib/apiis.xmllib	165
7.6	APIIS_HOME/lib/hierFrames.js	165
7.7	APIIS_HOME/lib/general.js	165
7.8	APIIS_HOME/lib/example_menu.js	165
7.9	APIIS_HOME/lib/formajax.js_neu	165
7.10	APIIS_HOME/lib/form_ulib.pm	165

7.11	APIIS_HOME/lib/navigation.js	166
7.12	APIIS_HOME/lib/yaform.pm	166
7.13	APIIS_HOME/lib/apiis2javascript.js	168
7.14	APIIS_HOME/lib/ref_breedprg_alib.pm	168
7.15	APIIS_HOME/lib/formajax.js	170
7.16	APIIS_HOME/lib/navigation.js	170
7.17	APIIS_HOME/lib/formneu.js	170
7.18	APIIS_HOME/lib/apiis2css.js	170
7.19	APIIS_HOME/lib/json.js	170
7.20	APIIS_HOME/lib/formnavigation.js	170
7.21	APIIS_HOME/lib/formajax.js.old	170
7.22	APIIS_HOME/lib/alt.js	170
7.23	APIIS_HOME/lib/Apiis/Load1.pm.org	170
7.24	APIIS_HOME/lib/Apiis/DataBase/Init.pm	170
7.25	APIIS_HOME/lib/Apiis/DataBase/Record.pm	171
7.26	APIIS_HOME/lib/Apiis/DataBase/User.pm	172
7.27	APIIS_HOME/lib/Apiis/DataBase/Record.pm_	173
7.28	APIIS_HOME/lib/Apiis/DataBase/SQL/MakeSQL.pm	174
7.29	APIIS_HOME/lib/Apiis/DataBase/SQL/sql_flow_example.pl	174
7.30	APIIS_HOME/lib/Apiis/DataBase/SQL/DirectSQL.pm	174
7.31	APIIS_HOME/lib/Apiis/DataBase/Record/Update.pm	175
7.32	APIIS_HOME/lib/Apiis/DataBase/Record/Modify.pm	175
7.33	APIIS_HOME/lib/Apiis/DataBase/Record/Insert.pm	175
7.34	APIIS_HOME/lib/Apiis/DataBase/Record/Check.pm	175
7.35	APIIS_HOME/lib/Apiis/DataBase/Record/Trigger.pm	175
7.36	APIIS_HOME/lib/Apiis/DataBase/Record/Column.pm	175
7.37	APIIS_HOME/lib/Apiis/DataBase/Record/Check/LastAction.pm	176
7.38	APIIS_HOME/lib/Apiis/DataBase/Record/Trigger.pm_	176
7.39	APIIS_HOME/lib/Apiis/DataBase/Record/Delete.pm	176
7.40	APIIS_HOME/lib/Apiis/Init.pm	177
7.41	APIIS_HOME/lib/Apiis/Report/Init.pm	177
7.42	APIIS_HOME/lib/Apiis/Report/InitXML.pm	177
7.43	APIIS_HOME/lib/Apiis/Report/Base.pm	178
7.44	APIIS_HOME/lib/Apiis/Report/PDF.pm	178
7.45	APIIS_HOME/lib/Apiis/Report/HTML.pm	178
7.46	APIIS_HOME/lib/Apiis/GUI.pm	179
7.47	APIIS_HOME/lib/Apiis/Load3.pm	179
7.48	APIIS_HOME/lib/Apiis/I18N/test.l10n	180
7.49	APIIS_HOME/lib/Apiis/Model.pm	180
7.50	APIIS_HOME/lib/Apiis/.Load3.pm.swp_	181
7.51	APIIS_HOME/lib/Apiis/Load.pm.org	181
7.52	APIIS_HOME/lib/Apiis/Load1.pm	182
7.53	APIIS_HOME/lib/Apiis/Auth/AR_View.pm	182
7.54	APIIS_HOME/lib/Apiis/Auth/AR_Auth.pm	182
7.55	APIIS_HOME/lib/Apiis/Auth/test.AR_Auth	182
7.85	APIIS_HOME/lib/Apiis/Auth/AR_ww/lib/languages.dat	184
7.86	APIIS_HOME/lib/Apiis/Auth/AR_ww/lib/HandleAJAX.pm	184
7.114	APIIS_HOME/lib/Apiis/Errors.pm	186
7.115	APIIS_HOME/lib/Apiis/Extjs/CHANGES_ext-3.1.0.html	186
7.116	APIIS_HOME/lib/Apiis/Extjs/gpl-3.0.txt	186

7.117	APIIS_HOME/lib/Apiis/Extjs/docs/output/tree.js	186
7.118	APIIS_HOME/lib/Apiis/Extjs/docs/resources/welcome.css . . .	186
7.119	APIIS_HOME/lib/Apiis/Extjs/docs/resources/favicon.ico . . .	186
7.120	APIIS_HOME/lib/Apiis/Extjs/docs/resources/style.css	186
7.121	APIIS_HOME/lib/Apiis/Extjs/docs/resources/prettify/prettify.css	186
7.122	APIIS_HOME/lib/Apiis/Extjs/docs/resources/prettify/prettify.js	186
7.123	APIIS_HOME/lib/Apiis/Extjs/docs/resources/wel-bg.gif	186
7.124	APIIS_HOME/lib/Apiis/Extjs/docs/resources/member-collapsed.gif	186
7.125	APIIS_HOME/lib/Apiis/Extjs/docs/resources/tree-bg.gif	186
7.126	APIIS_HOME/lib/Apiis/Extjs/docs/resources/expand-members.gif	186
7.127	APIIS_HOME/lib/Apiis/Extjs/docs/resources/member-expanded.gif	186
7.128	APIIS_HOME/lib/Apiis/Extjs/docs/resources/block-bg.gif . . .	186
7.129	APIIS_HOME/lib/Apiis/Extjs/docs/resources/expand-bg-over.gif	186
7.130	APIIS_HOME/lib/Apiis/Extjs/docs/resources/reset.css	186
7.131	APIIS_HOME/lib/Apiis/Extjs/docs/resources/expand-all.gif . .	186
7.132	APIIS_HOME/lib/Apiis/Extjs/docs/resources/elbow-end.gif . .	186
7.133	APIIS_HOME/lib/Apiis/Extjs/docs/resources/collapser.css . . .	186
7.134	APIIS_HOME/lib/Apiis/Extjs/docs/resources/icon-grid.gif . . .	186
7.135	APIIS_HOME/lib/Apiis/Extjs/docs/resources/print.css	186
7.136	APIIS_HOME/lib/Apiis/Extjs/docs/resources/member-hover.gif	186
7.137	APIIS_HOME/lib/Apiis/Extjs/docs/resources/hd-bg.gif	186
7.138	APIIS_HOME/lib/Apiis/Extjs/docs/resources/docs.css	186
7.139	APIIS_HOME/lib/Apiis/Extjs/docs/resources/TabCloseMenu.js	186
7.140	APIIS_HOME/lib/Apiis/Extjs/docs/resources/expand-bg.gif . .	186
7.141	APIIS_HOME/lib/Apiis/Extjs/docs/resources/ext-all.js	186
7.142	APIIS_HOME/lib/Apiis/Extjs/docs/resources/docs.js	186
7.143	APIIS_HOME/lib/Apiis/Extjs/docs/resources/form.png	186
7.144	APIIS_HOME/lib/Apiis/Extjs/docs/resources/collapse-all.gif .	186
7.145	APIIS_HOME/lib/Apiis/Extjs/docs/resources/css/ext-all.css . .	186
7.146	APIIS_HOME/lib/Apiis/Extjs/docs/resources/collapse-bg.gif .	186
7.147	APIIS_HOME/lib/Apiis/Extjs/docs/resources/block-bottom.gif	186
7.148	APIIS_HOME/lib/Apiis/Extjs/docs/resources/block-top.gif . .	186
7.149	APIIS_HOME/lib/Apiis/Extjs/docs/resources/brick.png	186
7.150	APIIS_HOME/lib/Apiis/Extjs/docs/resources/pkg-closed.gif . .	186
7.151	APIIS_HOME/lib/Apiis/Extjs/docs/resources/ext-base.js	186
7.152	APIIS_HOME/lib/Apiis/Extjs/docs/resources/hide-inherited.gif	186
7.153	APIIS_HOME/lib/Apiis/Extjs/docs/resources/pkg-open.gif . . .	186
7.154	APIIS_HOME/lib/Apiis/Extjs/docs/resources/welcome-bg.gif .	186
7.155	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-lang-en.html . .	186
7.156	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-lang-da.html . .	186
7.157	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-lang-vn.html . .	186
7.158	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-lang-mk.html .	186
7.159	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-base-region.html	186
7.160	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-lang-zh_CN.html	186
7.161	APIIS_HOME/lib/Apiis/Extjs/docs/source/Element.insertion-more.html	186
7.162	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-lang-fr_CA.html	186
7.163	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-base-begin.html	186
7.164	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-lang-pt_BR.html	186
7.165	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-lang-sr.html . .	186
7.166	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-lang-gr.html . .	186

7.167	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-lang-el_GR.html	186
7.168	APIIS_HOME/lib/Apiis/Extjs/docs/source/Ext-more.html . . .	186
7.169	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-lang-no_NN.html	186
7.170	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-lang-sl.html . .	186
7.171	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-lang-pt_PT.html	186
7.172	APIIS_HOME/lib/Apiis/Extjs/docs/source/Element.position-more.html	186
7.173	APIIS_HOME/lib/Apiis/Extjs/docs/source/prototype-bridge.html	186
7.174	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-lang-ro.html . .	186
7.175	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-base-anim-extra.html	186
7.176	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-lang-pl.html . .	186
7.177	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-lang-en_GB.html	186
7.178	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-lang-pt.html . .	186
7.179	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-base-end.html .	186
7.180	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-lang-lt.html . .	186
7.181	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-base-event.html	186
7.182	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-lang-cs.html . .	186
7.183	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-lang-ru.html . .	186
7.184	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-base-dom.html .	186
7.185	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-lang-tr.html . .	186
7.186	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-base-dom-more.html	186
7.187	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-lang-sr_RS.html	186
7.188	APIIS_HOME/lib/Apiis/Extjs/docs/source/Element.scroll-more.html	186
7.189	APIIS_HOME/lib/Apiis/Extjs/docs/source/jquery-bridge.html	186
7.190	APIIS_HOME/lib/Apiis/Extjs/docs/source/Observable-more.html	186
7.191	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-lang-nl.html . .	186
7.192	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-base-ajax.html .	186
7.193	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-lang-no_NB.html	186
7.194	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-lang-it.html . .	186
7.195	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-lang-fa.html . .	186
7.196	APIIS_HOME/lib/Apiis/Extjs/docs/source/yui-bridge.html . .	186
7.197	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-lang-hu.html . .	186
7.198	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-lang-de.html . .	186
7.199	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-base-point.html	186
7.200	APIIS_HOME/lib/Apiis/Extjs/docs/source/DomHelper-more.html	186
7.201	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-lang-bg.html . .	186
7.202	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-lang-ja.html . .	186
7.203	APIIS_HOME/lib/Apiis/Extjs/docs/source/Element.fx-more.html	186
7.204	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-lang-af.html . .	186
7.205	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-lang-hr.html . .	186
7.206	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-lang-fi.html . .	186
7.207	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-lang-ko.html . .	186
7.208	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-lang-th.html . .	186
7.209	APIIS_HOME/lib/Apiis/Extjs/docs/source/Element-more.html	186
7.210	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-lang-fr.html . .	186
7.211	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-lang-es.html . .	186
7.212	APIIS_HOME/lib/Apiis/Extjs/docs/source/EventManager-more.html	186
7.213	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-lang-zh_TW.html	186
7.214	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-lang-lv.html . .	186
7.215	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-lang-sv_SE.html	186
7.216	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-lang-he.html . .	186

7.217	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-lang-ca.html . .	186
7.218	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-lang-id.html . .	186
7.219	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-base-anim.html	186
7.220	APIIS_HOME/lib/Apiis/Extjs/docs/source/Element.style-more.html	186
7.221	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-lang-ukr.html .	186
7.222	APIIS_HOME/lib/Apiis/Extjs/docs/source/ext-lang-sk.html . .	186
7.223	APIIS_HOME/lib/Apiis/Extjs/docs/source/Element.traversal-more.html	186
7.224	APIIS_HOME/lib/Apiis/Extjs/docs/source/Template-more.html	186
7.225	APIIS_HOME/lib/Apiis/Extjs/docs/source/CompositeElementLite- more.html	186
7.226	APIIS_HOME/lib/Apiis/Extjs/license.txt	186
7.227	APIIS_HOME/lib/Apiis/Extjs/resources/expressinstall.swf . . .	186
7.228	APIIS_HOME/lib/Apiis/Extjs/resources/css/structure/grid.css	186
7.229	APIIS_HOME/lib/Apiis/Extjs/resources/css/structure/toolbar.css	186
7.230	APIIS_HOME/lib/Apiis/Extjs/resources/css/structure/layout.css	186
7.231	APIIS_HOME/lib/Apiis/Extjs/resources/css/structure/panel-reset.css	186
7.232	APIIS_HOME/lib/Apiis/Extjs/resources/css/structure/qtips.css	186
7.233	APIIS_HOME/lib/Apiis/Extjs/resources/css/structure/combo.css	186
7.234	APIIS_HOME/lib/Apiis/Extjs/resources/css/structure/reset.css	186
7.235	APIIS_HOME/lib/Apiis/Extjs/resources/css/structure/tabs.css	186
7.236	APIIS_HOME/lib/Apiis/Extjs/resources/css/structure/borders.css	186
7.237	APIIS_HOME/lib/Apiis/Extjs/resources/css/structure/debug.css	186
7.238	APIIS_HOME/lib/Apiis/Extjs/resources/css/structure/editor.css	186
7.239	APIIS_HOME/lib/Apiis/Extjs/resources/css/structure/date-picker.css	186
7.240	APIIS_HOME/lib/Apiis/Extjs/resources/css/structure/tree.css	186
7.241	APIIS_HOME/lib/Apiis/Extjs/resources/css/structure/form.css	186
7.242	APIIS_HOME/lib/Apiis/Extjs/resources/css/structure/core.css	186
7.243	APIIS_HOME/lib/Apiis/Extjs/resources/css/structure/list-view.css	186
7.244	APIIS_HOME/lib/Apiis/Extjs/resources/css/structure/box.css .	186
7.245	APIIS_HOME/lib/Apiis/Extjs/resources/css/structure/dialog.css	186
7.246	APIIS_HOME/lib/Apiis/Extjs/resources/css/structure/progress.css	186
7.247	APIIS_HOME/lib/Apiis/Extjs/resources/css/structure/slider.css	186
7.248	APIIS_HOME/lib/Apiis/Extjs/resources/css/structure/window.css	186
7.249	APIIS_HOME/lib/Apiis/Extjs/resources/css/structure/menu.css	186
7.250	APIIS_HOME/lib/Apiis/Extjs/resources/css/structure/resizable.css	186
7.251	APIIS_HOME/lib/Apiis/Extjs/resources/css/structure/button.css	186
7.252	APIIS_HOME/lib/Apiis/Extjs/resources/css/structure/panel.css	186
7.253	APIIS_HOME/lib/Apiis/Extjs/resources/css/structure/dd.css .	186
7.254	APIIS_HOME/lib/Apiis/Extjs/resources/css/visual/grid.css . .	186
7.255	APIIS_HOME/lib/Apiis/Extjs/resources/css/visual/toolbar.css	186
7.256	APIIS_HOME/lib/Apiis/Extjs/resources/css/visual/layout.css .	186
7.257	APIIS_HOME/lib/Apiis/Extjs/resources/css/visual/qtips.css . .	186
7.258	APIIS_HOME/lib/Apiis/Extjs/resources/css/visual/combo.css .	186
7.259	APIIS_HOME/lib/Apiis/Extjs/resources/css/visual/tabs.css . .	186
7.260	APIIS_HOME/lib/Apiis/Extjs/resources/css/visual/borders.css	186
7.261	APIIS_HOME/lib/Apiis/Extjs/resources/css/visual/debug.css .	186
7.262	APIIS_HOME/lib/Apiis/Extjs/resources/css/visual/editor.css .	186
7.263	APIIS_HOME/lib/Apiis/Extjs/resources/css/visual/date-picker.css	186
7.264	APIIS_HOME/lib/Apiis/Extjs/resources/css/visual/tree.css . .	186
7.265	APIIS_HOME/lib/Apiis/Extjs/resources/css/visual/form.css . .	186

7.266	APIIS_HOME/lib/Apiis/Extjs/resources/css/visual/core.css . .	186
7.267	APIIS_HOME/lib/Apiis/Extjs/resources/css/visual/list-view.css	186
7.268	APIIS_HOME/lib/Apiis/Extjs/resources/css/visual/box.css . .	186
7.269	APIIS_HOME/lib/Apiis/Extjs/resources/css/visual/dialog.css .	186
7.270	APIIS_HOME/lib/Apiis/Extjs/resources/css/visual/progress.css	186
7.271	APIIS_HOME/lib/Apiis/Extjs/resources/css/visual/slider.css .	186
7.272	APIIS_HOME/lib/Apiis/Extjs/resources/css/visual/window.css	186
7.273	APIIS_HOME/lib/Apiis/Extjs/resources/css/visual/menu.css .	186
7.274	APIIS_HOME/lib/Apiis/Extjs/resources/css/visual/resizable.css	186
7.275	APIIS_HOME/lib/Apiis/Extjs/resources/css/visual/button.css .	186
7.276	APIIS_HOME/lib/Apiis/Extjs/resources/css/visual/panel.css .	186
7.277	APIIS_HOME/lib/Apiis/Extjs/resources/css/visual/dd.css . . .	186
7.278	APIIS_HOME/lib/Apiis/Extjs/resources/css/ext-all.css	186
7.279	APIIS_HOME/lib/Apiis/Extjs/resources/css/debug.css	186
7.280	APIIS_HOME/lib/Apiis/Extjs/resources/css/xtheme-blue.css .	186
7.281	APIIS_HOME/lib/Apiis/Extjs/resources/css/xtheme-gray.css .	186
7.282	APIIS_HOME/lib/Apiis/Extjs/resources/css/README.txt . .	186
7.283	APIIS_HOME/lib/Apiis/Extjs/resources/css/reset-min.css . . .	186
7.284	APIIS_HOME/lib/Apiis/Extjs/resources/css/ext-all-notheme.css	186
7.285	APIIS_HOME/lib/Apiis/Extjs/resources/resources.jsb	186
7.286	APIIS_HOME/lib/Apiis/Extjs/resources/charts.swf	186
7.287	APIIS_HOME/lib/Apiis/Extjs/INCLUDE_ORDER.txt	186
7.288	APIIS_HOME/lib/Apiis/Extjs/ext-all-debug.js	186
7.289	APIIS_HOME/lib/Apiis/Extjs/adapters/jquery/jquery-adapter- debug.js	186
7.290	APIIS_HOME/lib/Apiis/Extjs/adapters/jquery/jquery-adapter.js	186
7.291	APIIS_HOME/lib/Apiis/Extjs/adapters/ext/ext-base-debug-w-comments.js	186
7.292	APIIS_HOME/lib/Apiis/Extjs/adapters/ext/ext-base-debug.js .	186
7.293	APIIS_HOME/lib/Apiis/Extjs/adapters/ext/ext-base.js	186
7.294	APIIS_HOME/lib/Apiis/Extjs/adapters/yui/ext-yui-adapter-debug.js	186
7.295	APIIS_HOME/lib/Apiis/Extjs/adapters/yui/ext-yui-adapter.js .	186
7.296	APIIS_HOME/lib/Apiis/Extjs/adapters/prototype/ext-prototype- adapter-debug.js	186
7.297	APIIS_HOME/lib/Apiis/Extjs/adapters/prototype/ext-prototype- adapter.js	186
7.298	APIIS_HOME/lib/Apiis/Extjs/ext-all.js	186
7.299	APIIS_HOME/lib/Apiis/Extjs/pkgs/pkg-history-debug.js	186
7.300	APIIS_HOME/lib/Apiis/Extjs/pkgs/pkg-forms-debug.js	186
7.301	APIIS_HOME/lib/Apiis/Extjs/pkgs/pkg-grid-foundation-debug.js	186
7.302	APIIS_HOME/lib/Apiis/Extjs/pkgs/pkg-grid-foundation.js . . .	186
7.303	APIIS_HOME/lib/Apiis/Extjs/pkgs/state.js	186
7.304	APIIS_HOME/lib/Apiis/Extjs/pkgs/pkg-buttons.js	186
7.305	APIIS_HOME/lib/Apiis/Extjs/pkgs/pkg-forms.js	186
7.306	APIIS_HOME/lib/Apiis/Extjs/pkgs/pkg-tree.js	186
7.307	APIIS_HOME/lib/Apiis/Extjs/pkgs/pkg-charts-debug.js	186
7.308	APIIS_HOME/lib/Apiis/Extjs/pkgs/pkg-menu.js	186
7.309	APIIS_HOME/lib/Apiis/Extjs/pkgs/pkg-tabs.js	186
7.310	APIIS_HOME/lib/Apiis/Extjs/pkgs/cmp-foundation-debug.js .	186
7.311	APIIS_HOME/lib/Apiis/Extjs/pkgs/pkg-buttons-debug.js . . .	186
7.312	APIIS_HOME/lib/Apiis/Extjs/pkgs/pkg-toolbars-debug.js . . .	186

7.313	APIIS_HOME/lib/Apiis/Extjs/pkgs/data-grouping.js	186
7.314	APIIS_HOME/lib/Apiis/Extjs/pkgs/direct-debug.js	186
7.315	APIIS_HOME/lib/Apiis/Extjs/pkgs/pkg-grid-grouping.js	186
7.316	APIIS_HOME/lib/Apiis/Extjs/pkgs/pkg-grid-property-debug.js	186
7.317	APIIS_HOME/lib/Apiis/Extjs/pkgs/cmp-foundation.js	186
7.318	APIIS_HOME/lib/Apiis/Extjs/pkgs/resizable.js	186
7.319	APIIS_HOME/lib/Apiis/Extjs/pkgs/data-json-debug.js	186
7.320	APIIS_HOME/lib/Apiis/Extjs/pkgs/window-debug.js	186
7.321	APIIS_HOME/lib/Apiis/Extjs/pkgs/data-foundation-debug.js	186
7.322	APIIS_HOME/lib/Apiis/Extjs/pkgs/window.js	186
7.323	APIIS_HOME/lib/Apiis/Extjs/pkgs/ext-foundation-debug.js	186
7.324	APIIS_HOME/lib/Apiis/Extjs/pkgs/pkg-grid-property.js	186
7.325	APIIS_HOME/lib/Apiis/Extjs/pkgs/pkg-tree-debug.js	186
7.326	APIIS_HOME/lib/Apiis/Extjs/pkgs/pkg-grid-editor.js	186
7.327	APIIS_HOME/lib/Apiis/Extjs/pkgs/state-debug.js	186
7.328	APIIS_HOME/lib/Apiis/Extjs/pkgs/pkg-tabs-debug.js	186
7.329	APIIS_HOME/lib/Apiis/Extjs/pkgs/data-list-views-debug.js	186
7.330	APIIS_HOME/lib/Apiis/Extjs/pkgs/pkg-tips.js	186
7.331	APIIS_HOME/lib/Apiis/Extjs/pkgs/ext-dd-debug.js	186
7.332	APIIS_HOME/lib/Apiis/Extjs/pkgs/data-grouping-debug.js	186
7.333	APIIS_HOME/lib/Apiis/Extjs/pkgs/data-xml.js	186
7.334	APIIS_HOME/lib/Apiis/Extjs/pkgs/pkg-history.js	186
7.335	APIIS_HOME/lib/Apiis/Extjs/pkgs/pkg-tips-debug.js	186
7.336	APIIS_HOME/lib/Apiis/Extjs/pkgs/pkg-grid-grouping-debug.js	186
7.337	APIIS_HOME/lib/Apiis/Extjs/pkgs/pkg-grid-editor-debug.js	186
7.338	APIIS_HOME/lib/Apiis/Extjs/pkgs/pkg-toolbars.js	186
7.339	APIIS_HOME/lib/Apiis/Extjs/pkgs/ext-foundation.js	186
7.340	APIIS_HOME/lib/Apiis/Extjs/pkgs/pkg-menu-debug.js	186
7.341	APIIS_HOME/lib/Apiis/Extjs/pkgs/data-list-views.js	186
7.342	APIIS_HOME/lib/Apiis/Extjs/pkgs/pkg-charts.js	186
7.343	APIIS_HOME/lib/Apiis/Extjs/pkgs/data-foundation.js	186
7.344	APIIS_HOME/lib/Apiis/Extjs/pkgs/ext-dd.js	186
7.345	APIIS_HOME/lib/Apiis/Extjs/pkgs/direct.js	186
7.346	APIIS_HOME/lib/Apiis/Extjs/pkgs/data-xml-debug.js	186
7.347	APIIS_HOME/lib/Apiis/Extjs/pkgs/data-json.js	186
7.348	APIIS_HOME/lib/Apiis/Extjs/pkgs/resizable-debug.js	186
7.349	APIIS_HOME/lib/Apiis/Extjs/examples/simple-widgets/progress- bar.js	186
7.350	APIIS_HOME/lib/Apiis/Extjs/examples/simple-widgets/progress- bar.html	186
7.351	APIIS_HOME/lib/Apiis/Extjs/examples/simple-widgets/qtips.css	186
7.352	APIIS_HOME/lib/Apiis/Extjs/examples/simple-widgets/progress- bar.css	186
7.353	APIIS_HOME/lib/Apiis/Extjs/examples/simple-widgets/editor.js	186
7.354	APIIS_HOME/lib/Apiis/Extjs/examples/simple-widgets/qtips.js	186
7.355	APIIS_HOME/lib/Apiis/Extjs/examples/simple-widgets/ajax-tip.html	186
7.356	APIIS_HOME/lib/Apiis/Extjs/examples/menu/menus.js	186
7.357	APIIS_HOME/lib/Apiis/Extjs/examples/menu/menu-show.gif	186
7.358	APIIS_HOME/lib/Apiis/Extjs/examples/menu/preview.png	186
7.359	APIIS_HOME/lib/Apiis/Extjs/examples/menu/menus.css	186

7.360	APIIS_HOME/lib/Apiis/Extjs/examples/menu/actions.js . . .	186
7.361	APIIS_HOME/lib/Apiis/Extjs/examples/menu/list-items.gif . .	186
7.362	APIIS_HOME/lib/Apiis/Extjs/examples/portal/sample-grid.js .	186
7.363	APIIS_HOME/lib/Apiis/Extjs/examples/portal/sample.css . . .	186
7.364	APIIS_HOME/lib/Apiis/Extjs/examples/portal/portal.js	186
7.365	APIIS_HOME/lib/Apiis/Extjs/examples/data.js	186
7.366	APIIS_HOME/lib/Apiis/Extjs/examples/grid-filtering/grid-filter-local.js	186
7.367	APIIS_HOME/lib/Apiis/Extjs/examples/grid-filtering/grid-filter-local.html	186
7.368	APIIS_HOME/lib/Apiis/Extjs/examples/grid-filtering/grid-filter.php	186
7.369	APIIS_HOME/lib/Apiis/Extjs/examples/grid-filtering/grid-demo.sql	186
7.370	APIIS_HOME/lib/Apiis/Extjs/examples/grid-filtering/grid-filter.json	186
7.371	APIIS_HOME/lib/Apiis/Extjs/examples/samples.css	186
7.372	APIIS_HOME/lib/Apiis/Extjs/examples/tabs/tabs-adv.js . . .	186
7.373	APIIS_HOME/lib/Apiis/Extjs/examples/tabs/tab-scroller-menu.css	186
7.374	APIIS_HOME/lib/Apiis/Extjs/examples/tabs/ajax1.htm	186
7.375	APIIS_HOME/lib/Apiis/Extjs/examples/tabs/tab-scroller-menu.gif	186
7.376	APIIS_HOME/lib/Apiis/Extjs/examples/tabs/tab-scroller-menu.html	186
7.377	APIIS_HOME/lib/Apiis/Extjs/examples/tabs/tabs-example.js .	186
7.378	APIIS_HOME/lib/Apiis/Extjs/examples/tabs/tabs-adv.html . .	186
7.379	APIIS_HOME/lib/Apiis/Extjs/examples/tabs/tab-scroller-menu.js	186
7.380	APIIS_HOME/lib/Apiis/Extjs/examples/tabs/tabs-example.css	186
7.381	APIIS_HOME/lib/Apiis/Extjs/examples/tabs/ajax2.htm	186
7.382	APIIS_HOME/lib/Apiis/Extjs/examples/tasks/gears.js	186
7.383	APIIS_HOME/lib/Apiis/Extjs/examples/tasks/tasks.css	186
7.384	APIIS_HOME/lib/Apiis/Extjs/examples/tasks/tasks.js	186
7.385	APIIS_HOME/lib/Apiis/Extjs/examples/tasks/classes.js	186
7.386	APIIS_HOME/lib/Apiis/Extjs/examples/tasks/db/ext-air-db.js	186
7.387	APIIS_HOME/lib/Apiis/Extjs/examples/tasks/db/ext-db.js . .	186
7.388	APIIS_HOME/lib/Apiis/Extjs/examples/tasks/db/ext-gears-db.js	186
7.389	APIIS_HOME/lib/Apiis/Extjs/examples/locale/ContactForm-nl.js	186
7.390	APIIS_HOME/lib/Apiis/Extjs/examples/locale/multi-lang.js . .	186
7.391	APIIS_HOME/lib/Apiis/Extjs/examples/locale/ContactForm.js	186
7.392	APIIS_HOME/lib/Apiis/Extjs/examples/locale/languages.js . .	186
7.393	APIIS_HOME/lib/Apiis/Extjs/examples/locale/multi-lang.html	186
7.394	APIIS_HOME/lib/Apiis/Extjs/examples/locale/dutch-provinces.js	186
7.395	APIIS_HOME/lib/Apiis/Extjs/examples/locale/dutch-form.html	186
7.396	APIIS_HOME/lib/Apiis/Extjs/examples/direct/direct-tree.js . .	186
7.397	APIIS_HOME/lib/Apiis/Extjs/examples/direct/php/api.php . .	186
7.398	APIIS_HOME/lib/Apiis/Extjs/examples/direct/php/poll.php .	186
7.399	APIIS_HOME/lib/Apiis/Extjs/examples/direct/php/classes/TestAction.php	186
7.400	APIIS_HOME/lib/Apiis/Extjs/examples/direct/php/classes/Profile.php	186
7.401	APIIS_HOME/lib/Apiis/Extjs/examples/direct/php/config.php	186
7.402	APIIS_HOME/lib/Apiis/Extjs/examples/direct/php/router.php	186
7.403	APIIS_HOME/lib/Apiis/Extjs/examples/direct/direct-form.php	186
7.404	APIIS_HOME/lib/Apiis/Extjs/examples/direct/direct-tree.php	186
7.405	APIIS_HOME/lib/Apiis/Extjs/examples/direct/direct.php . . .	186
7.406	APIIS_HOME/lib/Apiis/Extjs/examples/direct/direct.js	186
7.407	APIIS_HOME/lib/Apiis/Extjs/examples/direct/direct-form.js .	186

7.408	APIIS_HOME/lib/Apiis/Extjs/examples/.DS_Store	186
7.409	APIIS_HOME/lib/Apiis/Extjs/examples/grid/from-markup.js .	186
7.410	APIIS_HOME/lib/Apiis/Extjs/examples/grid/from-markup.html	186
7.411	APIIS_HOME/lib/Apiis/Extjs/examples/grid/buffer.js	186
7.412	APIIS_HOME/lib/Apiis/Extjs/examples/grid/grouping.css . . .	186
7.413	APIIS_HOME/lib/Apiis/Extjs/examples/grid/binding-with-classes.html	186
7.414	APIIS_HOME/lib/Apiis/Extjs/examples/grid/row-editor.html .	186
7.415	APIIS_HOME/lib/Apiis/Extjs/examples/grid/edit-grid.html . .	186
7.416	APIIS_HOME/lib/Apiis/Extjs/examples/grid/row-editor-data.json	186
7.417	APIIS_HOME/lib/Apiis/Extjs/examples/grid/property-grid.html	186
7.418	APIIS_HOME/lib/Apiis/Extjs/examples/grid/grouping.js . . .	186
7.419	APIIS_HOME/lib/Apiis/Extjs/examples/grid/progress-bar-pager.js	186
7.420	APIIS_HOME/lib/Apiis/Extjs/examples/grid/row-editor.js . . .	186
7.421	APIIS_HOME/lib/Apiis/Extjs/examples/grid/paging.js	186
7.422	APIIS_HOME/lib/Apiis/Extjs/examples/grid/property-grid.js .	186
7.423	APIIS_HOME/lib/Apiis/Extjs/examples/grid/xml-grid.js	186
7.424	APIIS_HOME/lib/Apiis/Extjs/examples/grid/totals.js	186
7.425	APIIS_HOME/lib/Apiis/Extjs/examples/grid/locking-grid.html	186
7.426	APIIS_HOME/lib/Apiis/Extjs/examples/grid/column-header-group.js	186
7.427	APIIS_HOME/lib/Apiis/Extjs/examples/grid/progress-bar-pager.html	186
7.428	APIIS_HOME/lib/Apiis/Extjs/examples/grid/locking-grid.js . .	186
7.429	APIIS_HOME/lib/Apiis/Extjs/examples/grid/array-grid.js . . .	186
7.430	APIIS_HOME/lib/Apiis/Extjs/examples/grid/totals-hybrid.html	186
7.431	APIIS_HOME/lib/Apiis/Extjs/examples/grid/grid-examples.css	186
7.432	APIIS_HOME/lib/Apiis/Extjs/examples/grid/grid-plugins.html	186
7.433	APIIS_HOME/lib/Apiis/Extjs/examples/grid/binding-with-classes.js	186
7.434	APIIS_HOME/lib/Apiis/Extjs/examples/grid/totals-hybrid.js .	186
7.435	APIIS_HOME/lib/Apiis/Extjs/examples/grid/edit-grid.js	186
7.436	APIIS_HOME/lib/Apiis/Extjs/examples/grid/sliding-pager.html	186
7.437	APIIS_HOME/lib/Apiis/Extjs/examples/grid/array-grid.html .	186
7.438	APIIS_HOME/lib/Apiis/Extjs/examples/grid/xml-grid.html . .	186
7.439	APIIS_HOME/lib/Apiis/Extjs/examples/grid/grid-plugins.js . .	186
7.440	APIIS_HOME/lib/Apiis/Extjs/examples/grid/sliding-pager.js .	186
7.441	APIIS_HOME/lib/Apiis/Extjs/examples/grid/totals-hybrid.json	186
7.442	APIIS_HOME/lib/Apiis/Extjs/examples/grid/binding.js	186
7.443	APIIS_HOME/lib/Apiis/Extjs/examples/grid/gen-names.js . .	186
7.444	APIIS_HOME/lib/Apiis/Extjs/examples/feed-viewer/FeedGrid.js	186
7.445	APIIS_HOME/lib/Apiis/Extjs/examples/feed-viewer/FeedPanel.js	186
7.446	APIIS_HOME/lib/Apiis/Extjs/examples/feed-viewer/FeedWindow.js	186
7.447	APIIS_HOME/lib/Apiis/Extjs/examples/feed-viewer/FeedViewer.js	186
7.448	APIIS_HOME/lib/Apiis/Extjs/examples/feed-viewer/feed-viewer.css	186
7.449	APIIS_HOME/lib/Apiis/Extjs/examples/feed-viewer/feed-proxy.php	186
7.450	APIIS_HOME/lib/Apiis/Extjs/examples/feed-viewer/MainPanel.js	186
7.451	APIIS_HOME/lib/Apiis/Extjs/examples/writer/writer.js	186
7.452	APIIS_HOME/lib/Apiis/Extjs/examples/writer/UserGrid.js . .	186
7.453	APIIS_HOME/lib/Apiis/Extjs/examples/writer/UserForm.js . .	186
7.454	APIIS_HOME/lib/Apiis/Extjs/examples/writer/writer-thumb.gif	186
7.455	APIIS_HOME/lib/Apiis/Extjs/examples/writer/writer.css . . .	186
7.456	APIIS_HOME/lib/Apiis/Extjs/examples/writer/app.php	186
7.457	APIIS_HOME/lib/Apiis/Extjs/examples/writer/remote/init.php	186

7.458	APIIS_HOME/lib/Apiis/Extjs/examples/writer/remote/app/controllers/users.php	186
7.459	APIIS_HOME/lib/Apiis/Extjs/examples/writer/remote/app/models/user.php	186
7.460	APIIS_HOME/lib/Apiis/Extjs/examples/writer/remote/lib/application_controller.php	186
7.461	APIIS_HOME/lib/Apiis/Extjs/examples/writer/remote/lib/request.php	186
7.462	APIIS_HOME/lib/Apiis/Extjs/examples/writer/remote/lib/model.php	186
7.463	APIIS_HOME/lib/Apiis/Extjs/examples/writer/remote/lib/session_db.php	186
7.464	APIIS_HOME/lib/Apiis/Extjs/examples/writer/remote/lib/response.php	186
7.465	APIIS_HOME/lib/Apiis/Extjs/examples/toolbar/toolbars.js	186
7.466	APIIS_HOME/lib/Apiis/Extjs/examples/toolbar/toolbars.css	186
7.467	APIIS_HOME/lib/Apiis/Extjs/examples/toolbar/overflow.js	186
7.468	APIIS_HOME/lib/Apiis/Extjs/examples/core/templates.js	186
7.469	APIIS_HOME/lib/Apiis/Extjs/examples/core/spotlight-example.js	186
7.470	APIIS_HOME/lib/Apiis/Extjs/examples/statusbar/statusbar-advanced.js	186
7.471	APIIS_HOME/lib/Apiis/Extjs/examples/statusbar/statusbar-demo.js	186
7.472	APIIS_HOME/lib/Apiis/Extjs/examples/statusbar/fake.php	186
7.473	APIIS_HOME/lib/Apiis/Extjs/examples/statusbar/statusbar-demo.html	186
7.474	APIIS_HOME/lib/Apiis/Extjs/examples/statusbar/statusbar-advanced.html	186
7.475	APIIS_HOME/lib/Apiis/Extjs/examples/ux/PortalColumn.js	186
7.476	APIIS_HOME/lib/Apiis/Extjs/examples/ux/MultiSelect.js	186
7.477	APIIS_HOME/lib/Apiis/Extjs/examples/ux/GMapPanel.js	186
7.478	APIIS_HOME/lib/Apiis/Extjs/examples/ux/LockingGridView.js	186
7.479	APIIS_HOME/lib/Apiis/Extjs/examples/ux/FieldReplicator.js	186
7.480	APIIS_HOME/lib/Apiis/Extjs/examples/ux/fileuploadfield/FileUploadField.js	186
7.481	APIIS_HOME/lib/Apiis/Extjs/examples/ux/fileuploadfield/css/fileuploadfield.css	186
7.482	APIIS_HOME/lib/Apiis/Extjs/examples/ux/SliderTip.js	186
7.483	APIIS_HOME/lib/Apiis/Extjs/examples/ux/SlidingPager.js	186
7.484	APIIS_HOME/lib/Apiis/Extjs/examples/ux/PanelResizer.js	186
7.485	APIIS_HOME/lib/Apiis/Extjs/examples/ux/Spotlight.js	186
7.486	APIIS_HOME/lib/Apiis/Extjs/examples/ux/ItemSelector.js	186
7.487	APIIS_HOME/lib/Apiis/Extjs/examples/ux/Spinner.js	186
7.488	APIIS_HOME/lib/Apiis/Extjs/examples/ux/statusbar/ValidationStatus.js	186
7.489	APIIS_HOME/lib/Apiis/Extjs/examples/ux/statusbar/StatusBar.js	186
7.490	APIIS_HOME/lib/Apiis/Extjs/examples/ux/statusbar/css/statusbar.css	186
7.491	APIIS_HOME/lib/Apiis/Extjs/examples/ux/GroupTabPanel.js	186
7.492	APIIS_HOME/lib/Apiis/Extjs/examples/ux/ColumnNodeUI.js	186
7.493	APIIS_HOME/lib/Apiis/Extjs/examples/ux/Focus.js	186
7.494	APIIS_HOME/lib/Apiis/Extjs/examples/ux/CheckColumn.js	186
7.495	APIIS_HOME/lib/Apiis/Extjs/examples/ux/TabScrollerMenu.js	186
7.496	APIIS_HOME/lib/Apiis/Extjs/examples/ux/SelectBox.js	186
7.497	APIIS_HOME/lib/Apiis/Extjs/examples/ux/ux-all.js	186
7.498	APIIS_HOME/lib/Apiis/Extjs/examples/ux/TabCloseMenu.js	186
7.499	APIIS_HOME/lib/Apiis/Extjs/examples/ux/RowEditor.js	186
7.500	APIIS_HOME/lib/Apiis/Extjs/examples/ux/ColumnHeaderGroup.js	186
7.501	APIIS_HOME/lib/Apiis/Extjs/examples/ux/GroupSummary.js	186
7.502	APIIS_HOME/lib/Apiis/Extjs/examples/ux/SpinnerField.js	186
7.503	APIIS_HOME/lib/Apiis/Extjs/examples/ux/Portal.js	186
7.504	APIIS_HOME/lib/Apiis/Extjs/examples/ux/treegrid/treegrid.css	186
7.505	APIIS_HOME/lib/Apiis/Extjs/examples/ux/treegrid/TreeGridLoader.js	186
7.506	APIIS_HOME/lib/Apiis/Extjs/examples/ux/treegrid/TreeGrid.js	186
7.507	APIIS_HOME/lib/Apiis/Extjs/examples/ux/treegrid/TreeGridColumns.js	186

7.508	APIIS_HOME/lib/Apiis/Extjs/examples/ux/treegrid/TreeGridSorter.js	186
7.509	APIIS_HOME/lib/Apiis/Extjs/examples/ux/treegrid/TreeGridNodeUI.js	186
7.510	APIIS_HOME/lib/Apiis/Extjs/examples/ux/treegrid/TreeGridColumnResizer.js	186
7.511	APIIS_HOME/lib/Apiis/Extjs/examples/ux/DataView-more.js	186
7.512	APIIS_HOME/lib/Apiis/Extjs/examples/ux/css/MultiSelect.css	186
7.513	APIIS_HOME/lib/Apiis/Extjs/examples/ux/css/ux-all.css	186
7.514	APIIS_HOME/lib/Apiis/Extjs/examples/ux/css/CenterLayout.css	186
7.515	APIIS_HOME/lib/Apiis/Extjs/examples/ux/css/GroupTab.css	186
7.516	APIIS_HOME/lib/Apiis/Extjs/examples/ux/css/Spinner.css	186
7.517	APIIS_HOME/lib/Apiis/Extjs/examples/ux/css/ColumnHeaderGroup.css	186
7.518	APIIS_HOME/lib/Apiis/Extjs/examples/ux/css/PanelResizer.css	186
7.519	APIIS_HOME/lib/Apiis/Extjs/examples/ux/css/Portal.css	186
7.520	APIIS_HOME/lib/Apiis/Extjs/examples/ux/css/GroupSummary.css	186
7.521	APIIS_HOME/lib/Apiis/Extjs/examples/ux/css/ColumnNodeUI.css	186
7.522	APIIS_HOME/lib/Apiis/Extjs/examples/ux/css/LockingGridView.css	186
7.523	APIIS_HOME/lib/Apiis/Extjs/examples/ux/css/RowEditor.css	186
7.524	APIIS_HOME/lib/Apiis/Extjs/examples/ux/BufferView.js	186
7.525	APIIS_HOME/lib/Apiis/Extjs/examples/ux/ux-all-debug.js	186
7.526	APIIS_HOME/lib/Apiis/Extjs/examples/ux/gridfilters/menu/RangeMenu.js	186
7.527	APIIS_HOME/lib/Apiis/Extjs/examples/ux/gridfilters/menu/ListMenu.js	186
7.528	APIIS_HOME/lib/Apiis/Extjs/examples/ux/gridfilters/filter/NumericFilter.js	186
7.529	APIIS_HOME/lib/Apiis/Extjs/examples/ux/gridfilters/filter/DateFilter.js	186
7.530	APIIS_HOME/lib/Apiis/Extjs/examples/ux/gridfilters/filter/ListFilter.js	186
7.531	APIIS_HOME/lib/Apiis/Extjs/examples/ux/gridfilters/filter/BooleanFilter.js	186
7.532	APIIS_HOME/lib/Apiis/Extjs/examples/ux/gridfilters/filter/StringFilter.js	186
7.533	APIIS_HOME/lib/Apiis/Extjs/examples/ux/gridfilters/filter/Filter.js	186
7.534	APIIS_HOME/lib/Apiis/Extjs/examples/ux/gridfilters/GridFilters.js	186
7.535	APIIS_HOME/lib/Apiis/Extjs/examples/ux/gridfilters/css/RangeMenu.css	186
7.536	APIIS_HOME/lib/Apiis/Extjs/examples/ux/gridfilters/css/GridFilters.css	186
7.537	APIIS_HOME/lib/Apiis/Extjs/examples/ux/SearchField.js	186
7.538	APIIS_HOME/lib/Apiis/Extjs/examples/ux/Portlet.js	186
7.539	APIIS_HOME/lib/Apiis/Extjs/examples/ux/CenterLayout.js	186
7.540	APIIS_HOME/lib/Apiis/Extjs/examples/ux/RowLayout.js	186
7.541	APIIS_HOME/lib/Apiis/Extjs/examples/ux/TableGrid.js	186
7.542	APIIS_HOME/lib/Apiis/Extjs/examples/ux/PagingMemoryProxy.js	186
7.543	APIIS_HOME/lib/Apiis/Extjs/examples/ux/FieldLabeler.js	186
7.544	APIIS_HOME/lib/Apiis/Extjs/examples/ux/RowExpander.js	186
7.545	APIIS_HOME/lib/Apiis/Extjs/examples/ux/GroupTab.js	186
7.546	APIIS_HOME/lib/Apiis/Extjs/examples/ux/XmlTreeLoader.js	186
7.547	APIIS_HOME/lib/Apiis/Extjs/examples/ux/ProgressBarPager.js	186
7.548	APIIS_HOME/lib/Apiis/Extjs/examples/desktop/credits.txt	186
7.549	APIIS_HOME/lib/Apiis/Extjs/examples/desktop/sample.js	186
7.550	APIIS_HOME/lib/Apiis/Extjs/examples/desktop/css/desktop.css	186
7.551	APIIS_HOME/lib/Apiis/Extjs/examples/desktop/js/Module.js	186
7.552	APIIS_HOME/lib/Apiis/Extjs/examples/desktop/js/StartMenu.js	186
7.553	APIIS_HOME/lib/Apiis/Extjs/examples/desktop/js/Desktop.js	186
7.554	APIIS_HOME/lib/Apiis/Extjs/examples/desktop/js/TaskBar.js	186
7.555	APIIS_HOME/lib/Apiis/Extjs/examples/desktop/js/App.js	186
7.556	APIIS_HOME/lib/Apiis/Extjs/examples/chart/charts.js	186
7.557	APIIS_HOME/lib/Apiis/Extjs/examples/chart/stacked-bar-chart.html	186

7.558	APIIS_HOME/lib/Apiis/Extjs/examples/chart/reload-chart.html	186
7.559	APIIS_HOME/lib/Apiis/Extjs/examples/chart/pie-chart.js . . .	186
7.560	APIIS_HOME/lib/Apiis/Extjs/examples/chart/reload-chart.js .	186
7.561	APIIS_HOME/lib/Apiis/Extjs/examples/chart/pie-chart.html .	186
7.562	APIIS_HOME/lib/Apiis/Extjs/examples/chart/stacked-bar-chart.js	186
7.563	APIIS_HOME/lib/Apiis/Extjs/examples/organizer/organizer.css	186
7.564	APIIS_HOME/lib/Apiis/Extjs/examples/organizer/organizer.js	186
7.565	APIIS_HOME/lib/Apiis/Extjs/examples/test-case-2b/resources/bubble- tb.gif	186
7.566	APIIS_HOME/lib/Apiis/Extjs/examples/test-case-2b/resources/bubble- fleedd.css	186
7.567	APIIS_HOME/lib/Apiis/Extjs/examples/test-case-2b/resources/bubble- lr.gif	186
7.568	APIIS_HOME/lib/Apiis/Extjs/examples/test-case-2b/resources/grid- configs.js	186
7.569	APIIS_HOME/lib/Apiis/Extjs/examples/test-case-2b/resources/form- configs.js	186
7.570	APIIS_HOME/lib/Apiis/Extjs/examples/test-case-2b/resources/bubble- cs-fleedd.gif	186
7.571	APIIS_HOME/lib/Apiis/Extjs/examples/test-case-2b/.DS_Store	186
7.572	APIIS_HOME/lib/Apiis/Extjs/examples/button/buttons.css . .	186
7.573	APIIS_HOME/lib/Apiis/Extjs/examples/button/buttons.js . .	186
7.574	APIIS_HOME/lib/Apiis/Extjs/examples/grouptabs/grouptabs.js	186
7.575	APIIS_HOME/lib/Apiis/Extjs/examples/treegrid/treegrid-data.json	186
7.576	APIIS_HOME/lib/Apiis/Extjs/examples/treegrid/tree-grid.js .	186
7.577	APIIS_HOME/lib/Apiis/Extjs/examples/forum/forum.css . . .	186
7.578	APIIS_HOME/lib/Apiis/Extjs/examples/forum/cmp-bg.gif . . .	186
7.579	APIIS_HOME/lib/Apiis/Extjs/examples/forum/message.png . .	186
7.580	APIIS_HOME/lib/Apiis/Extjs/examples/forum/message_edit.png	186
7.581	APIIS_HOME/lib/Apiis/Extjs/examples/forum/forum.js . . .	186
7.582	APIIS_HOME/lib/Apiis/Extjs/examples/forum/preview.png . .	186
7.583	APIIS_HOME/lib/Apiis/Extjs/examples/forum/messages.png .	186
7.584	APIIS_HOME/lib/Apiis/Extjs/examples/key-feed-viewer/FeedGrid.js	186
7.585	APIIS_HOME/lib/Apiis/Extjs/examples/key-feed-viewer/FeedPanel.js	186
7.586	APIIS_HOME/lib/Apiis/Extjs/examples/key-feed-viewer/FeedWindow.js	186
7.587	APIIS_HOME/lib/Apiis/Extjs/examples/key-feed-viewer/FeedViewer.js	186
7.588	APIIS_HOME/lib/Apiis/Extjs/examples/key-feed-viewer/feed-viewer.css	186
7.589	APIIS_HOME/lib/Apiis/Extjs/examples/key-feed-viewer/feed-proxy.php	186
7.590	APIIS_HOME/lib/Apiis/Extjs/examples/key-feed-viewer/MainPanel.js	186
7.591	APIIS_HOME/lib/Apiis/Extjs/examples/debug/debug.png . . .	186
7.592	APIIS_HOME/lib/Apiis/Extjs/examples/debug/debug-console.html	186
7.593	APIIS_HOME/lib/Apiis/Extjs/examples/multiselect/multiselect- demo.js	186
7.594	APIIS_HOME/lib/Apiis/Extjs/examples/multiselect/multiselect- demo.html	186
7.595	APIIS_HOME/lib/Apiis/Extjs/examples/tree/xml-tree-loader.css	186
7.596	APIIS_HOME/lib/Apiis/Extjs/examples/tree/ARIA.js	186
7.597	APIIS_HOME/lib/Apiis/Extjs/examples/tree/check-tree.html .	186
7.598	APIIS_HOME/lib/Apiis/Extjs/examples/tree/aria-tree.html . .	186
7.599	APIIS_HOME/lib/Apiis/Extjs/examples/tree/column-tree.js . .	186

7.600	APIIS_HOME/lib/Apiis/Extjs/examples/tree/check-tree.js . . .	186
7.601	APIIS_HOME/lib/Apiis/Extjs/examples/tree/column-tree.html	186
7.602	APIIS_HOME/lib/Apiis/Extjs/examples/tree/xml-tree-loader.html	186
7.603	APIIS_HOME/lib/Apiis/Extjs/examples/tree/two-trees.html . .	186
7.604	APIIS_HOME/lib/Apiis/Extjs/examples/tree/aria-tree.js	186
7.605	APIIS_HOME/lib/Apiis/Extjs/examples/tree/get-nodes.php . .	186
7.606	APIIS_HOME/lib/Apiis/Extjs/examples/tree/column-data.json	186
7.607	APIIS_HOME/lib/Apiis/Extjs/examples/tree/check-nodes.json	186
7.608	APIIS_HOME/lib/Apiis/Extjs/examples/tree/two-trees.js . . .	186
7.609	APIIS_HOME/lib/Apiis/Extjs/examples/tree/reorder.js	186
7.610	APIIS_HOME/lib/Apiis/Extjs/examples/tree/xml-tree-data.xml	186
7.611	APIIS_HOME/lib/Apiis/Extjs/examples/tree/column-tree.css .	186
7.612	APIIS_HOME/lib/Apiis/Extjs/examples/tree/save-dep.php . .	186
7.613	APIIS_HOME/lib/Apiis/Extjs/examples/tree/center-bg.gif . . .	186
7.614	APIIS_HOME/lib/Apiis/Extjs/examples/tree/xml-tree-loader.js	186
7.615	APIIS_HOME/lib/Apiis/Extjs/examples/message-box/msg-box.js	186
7.616	APIIS_HOME/lib/Apiis/Extjs/examples/message-box/msg-box.html	186
7.617	APIIS_HOME/lib/Apiis/Extjs/examples/spinner/spinner.js . .	186
7.618	APIIS_HOME/lib/Apiis/Extjs/examples/README.txt	186
7.619	APIIS_HOME/lib/Apiis/Extjs/examples/image-organizer/img-org.css	186
7.620	APIIS_HOME/lib/Apiis/Extjs/examples/image-organizer/readme.txt	186
7.621	APIIS_HOME/lib/Apiis/Extjs/examples/image-organizer/imgorg/ImageThumbPanel.js	186
7.622	APIIS_HOME/lib/Apiis/Extjs/examples/image-organizer/imgorg/MultiCombo.js	186
7.623	APIIS_HOME/lib/Apiis/Extjs/examples/image-organizer/imgorg/DirectCombo.js	186
7.624	APIIS_HOME/lib/Apiis/Extjs/examples/image-organizer/imgorg/TagWin.js	186
7.625	APIIS_HOME/lib/Apiis/Extjs/examples/image-organizer/imgorg/UploadQueue.js	186
7.626	APIIS_HOME/lib/Apiis/Extjs/examples/image-organizer/imgorg/ImageDv.js	186
7.627	APIIS_HOME/lib/Apiis/Extjs/examples/image-organizer/imgorg/Checkable.js	186
7.628	APIIS_HOME/lib/Apiis/Extjs/examples/image-organizer/imgorg/ImagePanel.js	186
7.629	APIIS_HOME/lib/Apiis/Extjs/examples/image-organizer/imgorg/AlbumsPanel.js	186
7.630	APIIS_HOME/lib/Apiis/Extjs/examples/image-organizer/imgorg/AlbumWin.js	186
7.631	APIIS_HOME/lib/Apiis/Extjs/examples/image-organizer/imgorg/AlbumTree.js	186
7.632	APIIS_HOME/lib/Apiis/Extjs/examples/image-organizer/imgorg/App.js	186
7.633	APIIS_HOME/lib/Apiis/Extjs/examples/image-organizer/php/api.php	186
7.634	APIIS_HOME/lib/Apiis/Extjs/examples/image-organizer/php/poll.php	186
7.635	APIIS_HOME/lib/Apiis/Extjs/examples/image-organizer/php/sql/setup.php	186
7.636	APIIS_HOME/lib/Apiis/Extjs/examples/image-organizer/php/classes/Tags.php	186
7.637	APIIS_HOME/lib/Apiis/Extjs/examples/image-organizer/php/classes/Images.php	186
7.638	APIIS_HOME/lib/Apiis/Extjs/examples/image-organizer/php/classes/Albums.php	186
7.639	APIIS_HOME/lib/Apiis/Extjs/examples/image-organizer/php/config.php	186
7.640	APIIS_HOME/lib/Apiis/Extjs/examples/image-organizer/php/router.php	186
7.641	APIIS_HOME/lib/Apiis/Extjs/examples/image-organizer/SWFUpload/swfupload.js	186
7.642	APIIS_HOME/lib/Apiis/Extjs/examples/image-organizer/SWFUpload/swfupload license.txt	186
7.643	APIIS_HOME/lib/Apiis/Extjs/examples/image-organizer/SWFUpload/plugins/swfupload.s	
7.644	APIIS_HOME/lib/Apiis/Extjs/examples/image-organizer/SWFUpload/plugins/SWFObject	
	License.txt	186
7.645	APIIS_HOME/lib/Apiis/Extjs/examples/image-organizer/SWFUpload/plugins/swfupload.q	
7.646	APIIS_HOME/lib/Apiis/Extjs/examples/image-organizer/SWFUpload/plugins/swfupload.c	
7.647	APIIS_HOME/lib/Apiis/Extjs/examples/image-organizer/SWFUpload/plugins/swfupload.s	

7.648	APIIS_HOME/lib/Apiis/Extjs/examples/image-organizer/SWFUpload/Flash/swfupload.swf	186
7.649	APIIS_HOME/lib/Apiis/Extjs/examples/history/history.js . . .	186
7.650	APIIS_HOME/lib/Apiis/Extjs/examples/window/layout.js . . .	186
7.651	APIIS_HOME/lib/Apiis/Extjs/examples/window/gmap.js . . .	186
7.652	APIIS_HOME/lib/Apiis/Extjs/examples/window/hello.js . . .	186
7.653	APIIS_HOME/lib/Apiis/Extjs/examples/layout-browser/layout-browser.html	186
7.654	APIIS_HOME/lib/Apiis/Extjs/examples/layout-browser/layouts/basic.js	186
7.655	APIIS_HOME/lib/Apiis/Extjs/examples/layout-browser/layouts/custom.js	186
7.656	APIIS_HOME/lib/Apiis/Extjs/examples/layout-browser/layouts/combination.js	186
7.657	APIIS_HOME/lib/Apiis/Extjs/examples/layout-browser/layout-browser.js	186
7.658	APIIS_HOME/lib/Apiis/Extjs/examples/layout-browser/layout-browser.css	186
7.659	APIIS_HOME/lib/Apiis/Extjs/examples/layout-browser/tree-data.json	186
7.660	APIIS_HOME/lib/Apiis/Extjs/examples/panel/bubble-panel.js	186
7.661	APIIS_HOME/lib/Apiis/Extjs/examples/panel/bubble-panel.html	186
7.662	APIIS_HOME/lib/Apiis/Extjs/examples/panel/css/bubble.css .	186
7.663	APIIS_HOME/lib/Apiis/Extjs/examples/panel/BubblePanel.js	186
7.664	APIIS_HOME/lib/Apiis/Extjs/examples/panel/panels.js	186
7.665	APIIS_HOME/lib/Apiis/Extjs/examples/dd/dnd_grid_to_formpanel.js	186
7.666	APIIS_HOME/lib/Apiis/Extjs/examples/dd/dnd_dropzones.js .	186
7.667	APIIS_HOME/lib/Apiis/Extjs/examples/dd/field-to-grid-dd.js .	186
7.668	APIIS_HOME/lib/Apiis/Extjs/examples/dd/dnd_grid_to_grid.js	186
7.669	APIIS_HOME/lib/Apiis/Extjs/examples/dd/field-to-grid-dd.html	186
7.670	APIIS_HOME/lib/Apiis/Extjs/examples/form/absform.js . . .	186
7.671	APIIS_HOME/lib/Apiis/Extjs/examples/form/states.js	186
7.672	APIIS_HOME/lib/Apiis/Extjs/examples/form/vbox-form.js . .	186
7.673	APIIS_HOME/lib/Apiis/Extjs/examples/form/forum-search.html	186
7.674	APIIS_HOME/lib/Apiis/Extjs/examples/form/check-radio.js . .	186
7.675	APIIS_HOME/lib/Apiis/Extjs/examples/form/file-upload.php .	186
7.676	APIIS_HOME/lib/Apiis/Extjs/examples/form/xml-form.js . . .	186
7.677	APIIS_HOME/lib/Apiis/Extjs/examples/form/adv-vtypes.html	186
7.678	APIIS_HOME/lib/Apiis/Extjs/examples/form/custom.js	186
7.679	APIIS_HOME/lib/Apiis/Extjs/examples/form/dynamic.js . . .	186
7.680	APIIS_HOME/lib/Apiis/Extjs/examples/form/anchoring.js . .	186
7.681	APIIS_HOME/lib/Apiis/Extjs/examples/form/check-radio.html	186
7.682	APIIS_HOME/lib/Apiis/Extjs/examples/form/file-upload.js . .	186
7.683	APIIS_HOME/lib/Apiis/Extjs/examples/form/file-upload.html	186
7.684	APIIS_HOME/lib/Apiis/Extjs/examples/form/form-grid.js . . .	186
7.685	APIIS_HOME/lib/Apiis/Extjs/examples/form/xml-form.xml . .	186
7.686	APIIS_HOME/lib/Apiis/Extjs/examples/form/xml-form.html .	186
7.687	APIIS_HOME/lib/Apiis/Extjs/examples/form/xml-errors.xml .	186
7.688	APIIS_HOME/lib/Apiis/Extjs/examples/form/adv-vtypes.js . .	186
7.689	APIIS_HOME/lib/Apiis/Extjs/examples/form/vbox-form.html .	186
7.690	APIIS_HOME/lib/Apiis/Extjs/examples/form/combos.js	186
7.691	APIIS_HOME/lib/Apiis/Extjs/examples/form/forms.css	186
7.692	APIIS_HOME/lib/Apiis/Extjs/examples/form/forum-search.js .	186
7.693	APIIS_HOME/lib/Apiis/Extjs/examples/form/combos.css . . .	186
7.694	APIIS_HOME/lib/Apiis/Extjs/examples/form/form-grid.html .	186

7.695	APIIS_HOME/lib/Apiis/Extjs/examples/restful/restful-thumb.gif	186
7.696	APIIS_HOME/lib/Apiis/Extjs/examples/restful/restful.js . . .	186
7.697	APIIS_HOME/lib/Apiis/Extjs/examples/restful/app.php . . .	186
7.698	APIIS_HOME/lib/Apiis/Extjs/examples/restful/restful.css . . .	186
7.699	APIIS_HOME/lib/Apiis/Extjs/examples/restful/remote/init.php	186
7.700	APIIS_HOME/lib/Apiis/Extjs/examples/restful/remote/app/controllers/users.php	186
7.701	APIIS_HOME/lib/Apiis/Extjs/examples/restful/remote/app/models/user.php	186
7.702	APIIS_HOME/lib/Apiis/Extjs/examples/restful/remote/lib/application_controller.php	186
7.703	APIIS_HOME/lib/Apiis/Extjs/examples/restful/remote/lib/request.php	186
7.704	APIIS_HOME/lib/Apiis/Extjs/examples/restful/remote/lib/model.php	186
7.705	APIIS_HOME/lib/Apiis/Extjs/examples/restful/remote/lib/session_db.php	186
7.706	APIIS_HOME/lib/Apiis/Extjs/examples/restful/remote/lib/response.php	186
7.707	APIIS_HOME/lib/Apiis/Extjs/examples/resizable/basic.js . . .	186
7.708	APIIS_HOME/lib/Apiis/Extjs/examples/resizable/basic.css . .	186
7.709	APIIS_HOME/lib/Apiis/Extjs/examples/view/list-view.js . . .	186
7.710	APIIS_HOME/lib/Apiis/Extjs/examples/view/get-images.php .	186
7.711	APIIS_HOME/lib/Apiis/Extjs/examples/view/data-view.js . . .	186
7.712	APIIS_HOME/lib/Apiis/Extjs/examples/view/chooser.js . . .	186
7.713	APIIS_HOME/lib/Apiis/Extjs/examples/view/data-view.html .	186
7.714	APIIS_HOME/lib/Apiis/Extjs/examples/view/list-view.css . . .	186
7.715	APIIS_HOME/lib/Apiis/Extjs/examples/view/chooser-example.js	186
7.716	APIIS_HOME/lib/Apiis/Extjs/examples/view/chooser.css . . .	186
7.717	APIIS_HOME/lib/Apiis/Extjs/examples/view/data-view.css . .	186
7.718	APIIS_HOME/lib/Apiis/Extjs/examples/view/list-view.html . .	186
7.719	APIIS_HOME/lib/Apiis/Extjs/examples/init.js	186
7.720	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/data-view.gif	186
7.721	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/form-xml.gif	186
7.722	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/layout- anchor.gif	186
7.723	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/grid-locking.gif	186
7.724	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/grid-summary.gif	186
7.725	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/toolbar- actions.gif	186
7.726	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/form-file- upload.gif	186
7.727	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/restful- thumb.gif	186
7.728	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/grid-plugins.gif	186
7.729	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/tree-columns.gif	186
7.730	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/list-view.gif	186
7.731	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/debug- console.gif	186
7.732	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/grid-data- binding.gif	186
7.733	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/tree-xml- loader.gif	186
7.734	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/dd-gridtoformpanel.gif	186
7.735	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/chart- pie.gif	186

7.736	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/layout-column.gif	186
7.737	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/form-multiselect.gif	186
7.738	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/progress-bar-pager.gif	186
7.739	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/locale-switch.gif	186
7.740	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/tree-reorder.gif	186
7.741	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/toolbar-overflow.gif	186
7.742	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/grid-row-editor.gif	186
7.743	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/window-layout.gif	186
7.744	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/form-check-radio.gif	186
7.745	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/grid-columngrouping.gif	186
7.746	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/statusbar-demo.gif	186
7.747	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/writer-thumb.gif	186
7.748	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/locale-dutch.gif	186
7.749	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/acc-tree.gif	186
7.750	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/form-adv-vtypes.gif	186
7.751	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/grid-grouping.gif	186
7.752	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/statusbar-adv.gif	186
7.753	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/grid-array.gif	186
7.754	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/form-vbox.gif	186
7.755	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/form-dynamic.gif	186
7.756	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/panel-bubble.gif	186
7.757	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/msg-box.gif	186
7.758	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/group-tabs.gif	186
7.759	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/grid-xml.gif	186
7.760	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/grid-buffer.gif	186
7.761	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/slider-pager.gif	186
7.762	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/dd-gridtogrid.gif	186
7.763	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/dd-zones.gif	186
7.764	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/tree-check.gif	186
7.765	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/layout-absolute.gif	186
7.766	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/gmap-panel.gif	186
7.767	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/tab-panel-scroller-menu.gif	186

7.768	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/layout-vbox.gif	186
7.769	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/grid-from-markup.gif	186
7.770	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/combo-custom.gif	186
7.771	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/border-layout.gif	186
7.772	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/grid-edit.gif	186
7.773	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/grid-filter.gif	186
7.774	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/layout-accordion.gif	186
7.775	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/form-custom.gif	186
7.776	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/layout-table.gif	186
7.777	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/grid-paging.gif	186
7.778	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/tabs-adv.gif	186
7.779	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/form-spinner.gif	186
7.780	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/chart-reload.gif	186
7.781	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/layout-form.gif	186
7.782	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/toolbar-button-groups.gif	186
7.783	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/layout-browser.gif	186
7.784	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/grid-property.gif	186
7.785	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/dd-fieldtogrid.gif	186
7.786	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/chart-stacked.gif	186
7.787	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/form-grid-binding.gif	186
7.788	APIIS_HOME/lib/Apiis/Extjs/examples/shared/screens/tree-two.gif	186
7.789	APIIS_HOME/lib/Apiis/Extjs/examples/shared/icons/fam/cog.png	186
7.790	APIIS_HOME/lib/Apiis/Extjs/examples/shared/icons/fam/cog_edit.png	186
7.791	APIIS_HOME/lib/Apiis/Extjs/examples/shared/icons/fam/application_view_list.png	186
7.792	APIIS_HOME/lib/Apiis/Extjs/examples/shared/icons/fam/user_edit.png	186
7.793	APIIS_HOME/lib/Apiis/Extjs/examples/shared/icons/fam/folder_wrench.png	186
7.794	APIIS_HOME/lib/Apiis/Extjs/examples/shared/icons/fam/application_go.png	186
7.795	APIIS_HOME/lib/Apiis/Extjs/examples/shared/icons/fam/user_female.png	186
7.796	APIIS_HOME/lib/Apiis/Extjs/examples/shared/icons/fam/user_red.png	186
7.797	APIIS_HOME/lib/Apiis/Extjs/examples/shared/icons/fam/image_add.png	186
7.798	APIIS_HOME/lib/Apiis/Extjs/examples/shared/icons/fam/rss_go.png	186
7.799	APIIS_HOME/lib/Apiis/Extjs/examples/shared/icons/fam/accept.png	186
7.800	APIIS_HOME/lib/Apiis/Extjs/examples/shared/icons/fam/user_add.png	186
7.801	APIIS_HOME/lib/Apiis/Extjs/examples/shared/icons/fam/information.png	186
7.802	APIIS_HOME/lib/Apiis/Extjs/examples/shared/icons/fam/user_delete.png	186
7.803	APIIS_HOME/lib/Apiis/Extjs/examples/shared/icons/fam/user.png	186
7.804	APIIS_HOME/lib/Apiis/Extjs/examples/shared/icons/fam/feed_add.png	186
7.805	APIIS_HOME/lib/Apiis/Extjs/examples/shared/icons/fam/connect.png	186

7.806APIIS_HOME/lib/Apiis/Extjs/examples/shared/icons/fam/feed_delete.png	186
7.807APIIS_HOME/lib/Apiis/Extjs/examples/shared/icons/fam/user_green.png	186
7.808APIIS_HOME/lib/Apiis/Extjs/examples/shared/icons/fam/error.png	186
7.809APIIS_HOME/lib/Apiis/Extjs/examples/shared/icons/fam/user_comment.png	186
7.810APIIS_HOME/lib/Apiis/Extjs/examples/shared/icons/fam/user_orange.png	186
7.811APIIS_HOME/lib/Apiis/Extjs/examples/shared/icons/fam/SILK.txt	186
7.812APIIS_HOME/lib/Apiis/Extjs/examples/shared/icons/fam/table_refresh.png	186
7.813APIIS_HOME/lib/Apiis/Extjs/examples/shared/icons/fam/feed_error.png	186
7.814APIIS_HOME/lib/Apiis/Extjs/examples/shared/icons/fam/user_gray.png	186
7.815APIIS_HOME/lib/Apiis/Extjs/examples/shared/icons/fam/book.png	186
7.816APIIS_HOME/lib/Apiis/Extjs/examples/shared/icons/fam/control_rewind.png	186
7.817APIIS_HOME/lib/Apiis/Extjs/examples/shared/icons/fam/user_suit.png	186
7.818APIIS_HOME/lib/Apiis/Extjs/examples/shared/icons/fam/add.png	186
7.819APIIS_HOME/lib/Apiis/Extjs/examples/shared/icons/fam/grid.png	186
7.820APIIS_HOME/lib/Apiis/Extjs/examples/shared/icons/fam/folder_go.png	186
7.821APIIS_HOME/lib/Apiis/Extjs/examples/shared/icons/arrow-up.gif	186
7.822APIIS_HOME/lib/Apiis/Extjs/examples/shared/icons/arrow-down.gif	186
7.823APIIS_HOME/lib/Apiis/Extjs/examples/shared/icons/silk.css	186
7.824APIIS_HOME/lib/Apiis/Extjs/examples/shared/lib-bar.png	186
7.825APIIS_HOME/lib/Apiis/Extjs/examples/shared/info-bg.gif	186
7.826APIIS_HOME/lib/Apiis/Extjs/examples/shared/examples.js	186
7.827APIIS_HOME/lib/Apiis/Extjs/examples/shared/examples.css	186
7.828APIIS_HOME/lib/Apiis/Extjs/examples/shared/extjs/ft.png	186
7.829APIIS_HOME/lib/Apiis/Extjs/examples/shared/extjs/topbar.png	186
7.830APIIS_HOME/lib/Apiis/Extjs/examples/shared/extjs/css/extjs.css	186
7.831APIIS_HOME/lib/Apiis/Extjs/examples/shared/extjs/site.js	186
7.832APIIS_HOME/lib/Apiis/Extjs/examples/shared/extjs/extjs2.png	186
7.833APIIS_HOME/lib/Apiis/Extjs/examples/shared/extjs/App.js	186
7.834APIIS_HOME/lib/Apiis/Extjs/examples/shared/code-display.js	186
7.835APIIS_HOME/lib/Apiis/Extjs/examples/state/save-state.php	186
7.836APIIS_HOME/lib/Apiis/Extjs/examples/state/SessionProvider.js	186
7.837APIIS_HOME/lib/Apiis/Extjs/examples/state/get-state.php	186
7.838APIIS_HOME/lib/Apiis/Extjs/examples/state/README.txt	186
7.839APIIS_HOME/lib/Apiis/Extjs/examples/slider/slider.js	186
7.840APIIS_HOME/lib/Apiis/Extjs/examples/slider/slider.css	186
7.841APIIS_HOME/lib/Apiis/Extjs/ext.jsb2	186
7.842APIIS_HOME/lib/Apiis/Extjs/src/widgets/menu/BaseItem.js	186
7.843APIIS_HOME/lib/Apiis/Extjs/src/widgets/menu/DateMenu.js	186
7.844APIIS_HOME/lib/Apiis/Extjs/src/widgets/menu/Separator.js	186
7.845APIIS_HOME/lib/Apiis/Extjs/src/widgets/menu/MenuMgr.js	186
7.846APIIS_HOME/lib/Apiis/Extjs/src/widgets/menu/Menu.js	186
7.847APIIS_HOME/lib/Apiis/Extjs/src/widgets/menu/ColorMenu.js	186
7.848APIIS_HOME/lib/Apiis/Extjs/src/widgets/menu/CheckItem.js	186
7.849APIIS_HOME/lib/Apiis/Extjs/src/widgets/menu/Item.js	186
7.850APIIS_HOME/lib/Apiis/Extjs/src/widgets/menu/TextItem.js	186
7.851APIIS_HOME/lib/Apiis/Extjs/src/widgets/Slider.js	186
7.852APIIS_HOME/lib/Apiis/Extjs/src/widgets/ColorPalette.js	186
7.853APIIS_HOME/lib/Apiis/Extjs/src/widgets/SplitButton.js	186
7.854APIIS_HOME/lib/Apiis/Extjs/src/widgets/Shadow.js	186
7.855APIIS_HOME/lib/Apiis/Extjs/src/widgets/Editor.js	186

7.856	APIIS_HOME/lib/Apiis/Extjs/src/widgets/PanelDD.js	186
7.857	APIIS_HOME/lib/Apiis/Extjs/src/widgets/WindowManager.js	186
7.858	APIIS_HOME/lib/Apiis/Extjs/src/widgets/PagingToolbar.js . .	186
7.859	APIIS_HOME/lib/Apiis/Extjs/src/widgets/Panel.js	186
7.860	APIIS_HOME/lib/Apiis/Extjs/src/widgets/tips/QuickTips.js .	186
7.861	APIIS_HOME/lib/Apiis/Extjs/src/widgets/tips/Tip.js	186
7.862	APIIS_HOME/lib/Apiis/Extjs/src/widgets/tips/QuickTip.js . .	186
7.863	APIIS_HOME/lib/Apiis/Extjs/src/widgets/tips/ToolTip.js . . .	186
7.864	APIIS_HOME/lib/Apiis/Extjs/src/widgets/ProgressBar.js . . .	186
7.865	APIIS_HOME/lib/Apiis/Extjs/src/widgets/BoxComponent.js .	186
7.866	APIIS_HOME/lib/Apiis/Extjs/src/widgets/grid/GridView.js . .	186
7.867	APIIS_HOME/lib/Apiis/Extjs/src/widgets/grid/ColumnModel.js	186
7.868	APIIS_HOME/lib/Apiis/Extjs/src/widgets/grid/PropertyGrid.js	186
7.869	APIIS_HOME/lib/Apiis/Extjs/src/widgets/grid/GroupingView.js	186
7.870	APIIS_HOME/lib/Apiis/Extjs/src/widgets/grid/GridPanel.js .	186
7.871	APIIS_HOME/lib/Apiis/Extjs/src/widgets/grid/EditorGrid.js .	186
7.872	APIIS_HOME/lib/Apiis/Extjs/src/widgets/grid/RowNumberer.js	186
7.873	APIIS_HOME/lib/Apiis/Extjs/src/widgets/grid/ColumnSplitDD.js	186
7.874	APIIS_HOME/lib/Apiis/Extjs/src/widgets/grid/RowSelectionModel.js	186
7.875	APIIS_HOME/lib/Apiis/Extjs/src/widgets/grid/GridDD.js . .	186
7.876	APIIS_HOME/lib/Apiis/Extjs/src/widgets/grid/GridEditor.js .	186
7.877	APIIS_HOME/lib/Apiis/Extjs/src/widgets/grid/CellSelectionModel.js	186
7.878	APIIS_HOME/lib/Apiis/Extjs/src/widgets/grid/AbstractSelectionModel.js	186
7.879	APIIS_HOME/lib/Apiis/Extjs/src/widgets/grid/Column.js . . .	186
7.880	APIIS_HOME/lib/Apiis/Extjs/src/widgets/grid/CheckboxSelectionModel.js	186
7.881	APIIS_HOME/lib/Apiis/Extjs/src/widgets/grid/ColumnDD.js .	186
7.882	APIIS_HOME/lib/Apiis/Extjs/src/widgets/list/Sorter.js	186
7.883	APIIS_HOME/lib/Apiis/Extjs/src/widgets/list/ColumnResizer.js	186
7.884	APIIS_HOME/lib/Apiis/Extjs/src/widgets/list/ListView.js . . .	186
7.885	APIIS_HOME/lib/Apiis/Extjs/src/widgets/list/Column.js . . .	186
7.886	APIIS_HOME/lib/Apiis/Extjs/src/widgets/Action.js	186
7.887	APIIS_HOME/lib/Apiis/Extjs/src/widgets/Resizable.js	186
7.888	APIIS_HOME/lib/Apiis/Extjs/src/widgets/chart/swfobject.js .	186
7.889	APIIS_HOME/lib/Apiis/Extjs/src/widgets/chart/FlashComponent.js	186
7.890	APIIS_HOME/lib/Apiis/Extjs/src/widgets/chart/Chart.js . . .	186
7.891	APIIS_HOME/lib/Apiis/Extjs/src/widgets/chart/EventProxy.js	186
7.892	APIIS_HOME/lib/Apiis/Extjs/src/widgets/Component.js . . .	186
7.893	APIIS_HOME/lib/Apiis/Extjs/src/widgets/ComponentMgr.js .	186
7.894	APIIS_HOME/lib/Apiis/Extjs/src/widgets/DataView.js	186
7.895	APIIS_HOME/lib/Apiis/Extjs/src/widgets/tree/TreeFilter.js .	186
7.896	APIIS_HOME/lib/Apiis/Extjs/src/widgets/tree/TreeDragZone.js	186
7.897	APIIS_HOME/lib/Apiis/Extjs/src/widgets/tree/TreeSorter.js .	186
7.898	APIIS_HOME/lib/Apiis/Extjs/src/widgets/tree/TreeNodeUI.js	186
7.899	APIIS_HOME/lib/Apiis/Extjs/src/widgets/tree/TreeNode.js . .	186
7.900	APIIS_HOME/lib/Apiis/Extjs/src/widgets/tree/TreePanel.js . .	186
7.901	APIIS_HOME/lib/Apiis/Extjs/src/widgets/tree/AsyncTreeNode.js	186
7.902	APIIS_HOME/lib/Apiis/Extjs/src/widgets/tree/TreeEventModel.js	186
7.903	APIIS_HOME/lib/Apiis/Extjs/src/widgets/tree/TreeDropZone.js	186
7.904	APIIS_HOME/lib/Apiis/Extjs/src/widgets/tree/TreeLoader.js .	186
7.905	APIIS_HOME/lib/Apiis/Extjs/src/widgets/tree/TreeEditor.js .	186

7.906APIIS_HOME/lib/Apiis/Extjs/src/widgets/tree/TreeSelectionModel.js	186
7.907APIIS_HOME/lib/Apiis/Extjs/src/widgets/Container.js	186
7.908APIIS_HOME/lib/Apiis/Extjs/src/widgets/Toolbar.js	186
7.909APIIS_HOME/lib/Apiis/Extjs/src/widgets/DatePicker.js	186
7.910APIIS_HOME/lib/Apiis/Extjs/src/widgets/layout/CardLayout.js	186
7.911APIIS_HOME/lib/Apiis/Extjs/src/widgets/layout/FitLayout.js	186
7.912APIIS_HOME/lib/Apiis/Extjs/src/widgets/layout/BorderLayout.js	186
7.913APIIS_HOME/lib/Apiis/Extjs/src/widgets/layout/AnchorLayout.js	186
7.914APIIS_HOME/lib/Apiis/Extjs/src/widgets/layout/FormLayout.js	186
7.915APIIS_HOME/lib/Apiis/Extjs/src/widgets/layout/ColumnLayout.js	186
7.916APIIS_HOME/lib/Apiis/Extjs/src/widgets/layout/BoxLayout.js	186
7.917APIIS_HOME/lib/Apiis/Extjs/src/widgets/layout/AccordionLayout.js	186
7.918APIIS_HOME/lib/Apiis/Extjs/src/widgets/layout/ContainerLayout.js	186
7.919APIIS_HOME/lib/Apiis/Extjs/src/widgets/layout/AbsoluteLayout.js	186
7.920APIIS_HOME/lib/Apiis/Extjs/src/widgets/layout/TableLayout.js	186
7.921APIIS_HOME/lib/Apiis/Extjs/src/widgets/TabPanel.js	186
7.922APIIS_HOME/lib/Apiis/Extjs/src/widgets/Layer.js	186
7.923APIIS_HOME/lib/Apiis/Extjs/src/widgets/Button.js	186
7.924APIIS_HOME/lib/Apiis/Extjs/src/widgets/CycleButton.js . . .	186
7.925APIIS_HOME/lib/Apiis/Extjs/src/widgets/Viewport.js	186
7.926APIIS_HOME/lib/Apiis/Extjs/src/widgets/Window.js	186
7.927APIIS_HOME/lib/Apiis/Extjs/src/widgets/LoadMask.js	186
7.928APIIS_HOME/lib/Apiis/Extjs/src/widgets/form/DateField.js .	186
7.929APIIS_HOME/lib/Apiis/Extjs/src/widgets/form/Label.js . . .	186
7.930APIIS_HOME/lib/Apiis/Extjs/src/widgets/form/Field.js . . .	186
7.931APIIS_HOME/lib/Apiis/Extjs/src/widgets/form/TriggerField.js	186
7.932APIIS_HOME/lib/Apiis/Extjs/src/widgets/form/RadioGroup.js	186
7.933APIIS_HOME/lib/Apiis/Extjs/src/widgets/form/CheckboxGroup.js	186
7.934APIIS_HOME/lib/Apiis/Extjs/src/widgets/form/DisplayField.js	186
7.935APIIS_HOME/lib/Apiis/Extjs/src/widgets/form/BasicForm.js .	186
7.936APIIS_HOME/lib/Apiis/Extjs/src/widgets/form/VTypes.js . .	186
7.937APIIS_HOME/lib/Apiis/Extjs/src/widgets/form/TimeField.js .	186
7.938APIIS_HOME/lib/Apiis/Extjs/src/widgets/form/Action.js . . .	186
7.939APIIS_HOME/lib/Apiis/Extjs/src/widgets/form/FieldSet.js . .	186
7.940APIIS_HOME/lib/Apiis/Extjs/src/widgets/form/HtmlEditor.js	186
7.941APIIS_HOME/lib/Apiis/Extjs/src/widgets/form/Combo.js . . .	186
7.942APIIS_HOME/lib/Apiis/Extjs/src/widgets/form/Form.js	186
7.943APIIS_HOME/lib/Apiis/Extjs/src/widgets/form/Hidden.js . . .	186
7.944APIIS_HOME/lib/Apiis/Extjs/src/widgets/form/Checkbox.js .	186
7.945APIIS_HOME/lib/Apiis/Extjs/src/widgets/form/Radio.js . . .	186
7.946APIIS_HOME/lib/Apiis/Extjs/src/widgets/form/TextArea.js .	186
7.947APIIS_HOME/lib/Apiis/Extjs/src/widgets/form/TextField.js .	186
7.948APIIS_HOME/lib/Apiis/Extjs/src/widgets/form/NumberField.js	186
7.949APIIS_HOME/lib/Apiis/Extjs/src/widgets/SplitBar.js	186
7.950APIIS_HOME/lib/Apiis/Extjs/src/widgets/ButtonGroup.js . .	186
7.951APIIS_HOME/lib/Apiis/Extjs/src/widgets/MessageBox.js . . .	186
7.952APIIS_HOME/lib/Apiis/Extjs/src/locale/ext-lang-it.js	186
7.953APIIS_HOME/lib/Apiis/Extjs/src/locale/ext-lang-nl.js	186
7.954APIIS_HOME/lib/Apiis/Extjs/src/locale/ext-lang-tr.js	186
7.955APIIS_HOME/lib/Apiis/Extjs/src/locale/ext-lang-ukr.js	186

7.956	APIIS_HOME/lib/Apiis/Extjs/src/locale/ext-lang-id.js	186
7.957	APIIS_HOME/lib/Apiis/Extjs/src/locale/ext-lang-fr_CA.js . . .	186
7.958	APIIS_HOME/lib/Apiis/Extjs/src/locale/ext-lang-fi.js	186
7.959	APIIS_HOME/lib/Apiis/Extjs/src/locale/ext-lang-fr.js	186
7.960	APIIS_HOME/lib/Apiis/Extjs/src/locale/ext-lang-el_GR.js . . .	186
7.961	APIIS_HOME/lib/Apiis/Extjs/src/locale/ext-lang-th.js	186
7.962	APIIS_HOME/lib/Apiis/Extjs/src/locale/ext-lang-sl.js	186
7.963	APIIS_HOME/lib/Apiis/Extjs/src/locale/ext-lang-pt.js	186
7.964	APIIS_HOME/lib/Apiis/Extjs/src/locale/ext-lang-cs.js	186
7.965	APIIS_HOME/lib/Apiis/Extjs/src/locale/ext-lang-pt_BR.js . . .	186
7.966	APIIS_HOME/lib/Apiis/Extjs/src/locale/ext-lang-bg.js	186
7.967	APIIS_HOME/lib/Apiis/Extjs/src/locale/ext-lang-sr.js	186
7.968	APIIS_HOME/lib/Apiis/Extjs/src/locale/ext-lang-sv_SE.js . . .	186
7.969	APIIS_HOME/lib/Apiis/Extjs/src/locale/ext-lang-pt_PT.js . . .	186
7.970	APIIS_HOME/lib/Apiis/Extjs/src/locale/ext-lang-ru.js	186
7.971	APIIS_HOME/lib/Apiis/Extjs/src/locale/ext-lang-ro.js	186
7.972	APIIS_HOME/lib/Apiis/Extjs/src/locale/ext-lang-mk.js	186
7.973	APIIS_HOME/lib/Apiis/Extjs/src/locale/ext-lang-en.js	186
7.974	APIIS_HOME/lib/Apiis/Extjs/src/locale/ext-lang-lt.js	186
7.975	APIIS_HOME/lib/Apiis/Extjs/src/locale/ext-lang-zh_TW.js . . .	186
7.976	APIIS_HOME/lib/Apiis/Extjs/src/locale/ext-lang-ja.js	186
7.977	APIIS_HOME/lib/Apiis/Extjs/src/locale/ext-lang-de.js	186
7.978	APIIS_HOME/lib/Apiis/Extjs/src/locale/ext-lang-hu.js	186
7.979	APIIS_HOME/lib/Apiis/Extjs/src/locale/ext-lang-sr_RS.js . . .	186
7.980	APIIS_HOME/lib/Apiis/Extjs/src/locale/ext-lang-af.js	186
7.981	APIIS_HOME/lib/Apiis/Extjs/src/locale/ext-lang-fa.js	186
7.982	APIIS_HOME/lib/Apiis/Extjs/src/locale/ext-lang-hr.js	186
7.983	APIIS_HOME/lib/Apiis/Extjs/src/locale/ext-lang-en_GB.js . . .	186
7.984	APIIS_HOME/lib/Apiis/Extjs/src/locale/ext-lang-zh_CN.js . . .	186
7.985	APIIS_HOME/lib/Apiis/Extjs/src/locale/ext-lang-es.js	186
7.986	APIIS_HOME/lib/Apiis/Extjs/src/locale/ext-lang-ko.js	186
7.987	APIIS_HOME/lib/Apiis/Extjs/src/locale/ext-lang-ca.js	186
7.988	APIIS_HOME/lib/Apiis/Extjs/src/locale/ext-lang-no_NB.js . . .	186
7.989	APIIS_HOME/lib/Apiis/Extjs/src/locale/ext-lang-pl.js	186
7.990	APIIS_HOME/lib/Apiis/Extjs/src/locale/ext-lang-vn.js	186
7.991	APIIS_HOME/lib/Apiis/Extjs/src/locale/ext-lang-no_NN.js . . .	186
7.992	APIIS_HOME/lib/Apiis/Extjs/src/locale/ext-lang-he.js	186
7.993	APIIS_HOME/lib/Apiis/Extjs/src/locale/ext-lang-da.js	186
7.994	APIIS_HOME/lib/Apiis/Extjs/src/locale/ext-lang-lv.js	186
7.995	APIIS_HOME/lib/Apiis/Extjs/src/locale/ext-lang-sk.js	186
7.996	APIIS_HOME/lib/Apiis/Extjs/src/locale/ext-lang-gr.js	186
7.997	APIIS_HOME/lib/Apiis/Extjs/src/direct/Provider.js	186
7.998	APIIS_HOME/lib/Apiis/Extjs/src/direct/RemotingProvider.js . .	186
7.999	APIIS_HOME/lib/Apiis/Extjs/src/direct/Direct.js	186
7.1000	APIIS_HOME/lib/Apiis/Extjs/src/direct/JsonProvider.js	186
7.1001	APIIS_HOME/lib/Apiis/Extjs/src/direct/Event.js	186
7.1002	APIIS_HOME/lib/Apiis/Extjs/src/direct/PollingProvider.js . .	186
7.1003	APIIS_HOME/lib/Apiis/Extjs/src/direct/Transaction.js	186
7.1004	APIIS_HOME/lib/Apiis/Extjs/src/.DS_Store	186
7.1005	APIIS_HOME/lib/Apiis/Extjs/src/adaptor/ext-base-dom-more.js	186

7.100	APIIS_HOME/lib/Apiis/Extjs/src/adaptor/prototype-bridge.js	186
7.100	APIIS_HOME/lib/Apiis/Extjs/src/adaptor/core/ext-base-ajax.js	186
7.100	APIIS_HOME/lib/Apiis/Extjs/src/adaptor/core/ext-base-end.js	186
7.100	APIIS_HOME/lib/Apiis/Extjs/src/adaptor/core/ext-base-anim.js	186
7.101	APIIS_HOME/lib/Apiis/Extjs/src/adaptor/core/ext-base-region.js	186
7.101	APIIS_HOME/lib/Apiis/Extjs/src/adaptor/core/ext-base-point.js	186
7.101	APIIS_HOME/lib/Apiis/Extjs/src/adaptor/core/ext-base-anim-extra.js	186
7.101	APIIS_HOME/lib/Apiis/Extjs/src/adaptor/core/ext-base-event.js	186
7.101	APIIS_HOME/lib/Apiis/Extjs/src/adaptor/core/ext-base-dom.js	186
7.101	APIIS_HOME/lib/Apiis/Extjs/src/adaptor/core/ext-base-begin.js	186
7.101	APIIS_HOME/lib/Apiis/Extjs/src/adaptor/yui-bridge.js	186
7.101	APIIS_HOME/lib/Apiis/Extjs/src/adaptor/jquery-bridge.js	186
7.101	APIIS_HOME/lib/Apiis/Extjs/src/core/Element.scroll-more.js	186
7.101	APIIS_HOME/lib/Apiis/Extjs/src/core/Element.traversal-more.js	186
7.102	APIIS_HOME/lib/Apiis/Extjs/src/core/Element.alignment.js	186
7.102	APIIS_HOME/lib/Apiis/Extjs/src/core/Ext-more.js	186
7.102	APIIS_HOME/lib/Apiis/Extjs/src/core/Element.legacy.js	186
7.102	APIIS_HOME/lib/Apiis/Extjs/src/core/CompositeElement.js	186
7.102	APIIS_HOME/lib/Apiis/Extjs/src/core/Element.style-more.js	186
7.102	APIIS_HOME/lib/Apiis/Extjs/src/core/CompositeElementLite-more.js	186
7.102	APIIS_HOME/lib/Apiis/Extjs/src/core/Error.js	186
7.102	APIIS_HOME/lib/Apiis/Extjs/src/core/core/DomHelper.js	186
7.102	APIIS_HOME/lib/Apiis/Extjs/src/core/core/EventManager.js	186
7.102	APIIS_HOME/lib/Apiis/Extjs/src/core/core/Element.style.js	186
7.103	APIIS_HOME/lib/Apiis/Extjs/src/core/core/Ext.js	186
7.103	APIIS_HOME/lib/Apiis/Extjs/src/core/core/Element.traversal.js	186
7.103	APIIS_HOME/lib/Apiis/Extjs/src/core/core/Fx.js	186
7.103	APIIS_HOME/lib/Apiis/Extjs/src/core/core/Element.insertion.js	186
7.103	APIIS_HOME/lib/Apiis/Extjs/src/core/core/Template.js	186
7.103	APIIS_HOME/lib/Apiis/Extjs/src/core/core/Element.position.js	186
7.103	APIIS_HOME/lib/Apiis/Extjs/src/core/core/Element.fx.js	186
7.103	APIIS_HOME/lib/Apiis/Extjs/src/core/core/DomQuery.js	186
7.103	APIIS_HOME/lib/Apiis/Extjs/src/core/core/Element.js	186
7.103	APIIS_HOME/lib/Apiis/Extjs/src/core/core/CompositeElementLite.js	186
7.104	APIIS_HOME/lib/Apiis/Extjs/src/core/core/Element.scroll.js	186
7.104	APIIS_HOME/lib/Apiis/Extjs/src/core/Element.keys.js	186
7.104	APIIS_HOME/lib/Apiis/Extjs/src/core/DomHelper-more.js	186
7.104	APIIS_HOME/lib/Apiis/Extjs/src/core/Template-more.js	186
7.104	APIIS_HOME/lib/Apiis/Extjs/src/core/EventManager-more.js	186
7.104	APIIS_HOME/lib/Apiis/Extjs/src/core/Element.insertion-more.js	186
7.104	APIIS_HOME/lib/Apiis/Extjs/src/core/Element.position-more.js	186
7.104	APIIS_HOME/lib/Apiis/Extjs/src/core/Element-more.js	186
7.104	APIIS_HOME/lib/Apiis/Extjs/src/core/Element.fx-more.js	186
7.104	APIIS_HOME/lib/Apiis/Extjs/src/core/Element.dd.js	186
7.105	APIIS_HOME/lib/Apiis/Extjs/src/data/XmlWriter.js	186
7.105	APIIS_HOME/lib/Apiis/Extjs/src/data/Request.js	186
7.105	APIIS_HOME/lib/Apiis/Extjs/src/data/XmlReader.js	186
7.105	APIIS_HOME/lib/Apiis/Extjs/src/data/DataReader.js	186

7.105	APIIS_HOME/lib/Apiis/Extjs/src/data/Record.js	186
7.105	APIIS_HOME/lib/Apiis/Extjs/src/data/DirectProxy.js	186
7.105	APIIS_HOME/lib/Apiis/Extjs/src/data/JsonReader.js	186
7.105	APIIS_HOME/lib/Apiis/Extjs/src/data/DataWriter.js	186
7.105	APIIS_HOME/lib/Apiis/Extjs/src/data/SortTypes.js	186
7.105	APIIS_HOME/lib/Apiis/Extjs/src/data/DataProxy.js	186
7.106	APIIS_HOME/lib/Apiis/Extjs/src/data/Tree.js	186
7.106	APIIS_HOME/lib/Apiis/Extjs/src/data/core/Connection.js	186
7.106	APIIS_HOME/lib/Apiis/Extjs/src/data/ArrayStore.js	186
7.106	APIIS_HOME/lib/Apiis/Extjs/src/data/XmlStore.js	186
7.106	APIIS_HOME/lib/Apiis/Extjs/src/data/DataField.js	186
7.106	APIIS_HOME/lib/Apiis/Extjs/src/data/DirectStore.js	186
7.106	APIIS_HOME/lib/Apiis/Extjs/src/data/Response.js	186
7.106	APIIS_HOME/lib/Apiis/Extjs/src/data/ArrayReader.js	186
7.106	APIIS_HOME/lib/Apiis/Extjs/src/data/JsonStore.js	186
7.106	APIIS_HOME/lib/Apiis/Extjs/src/data/ScriptTagProxy.js	186
7.107	APIIS_HOME/lib/Apiis/Extjs/src/data/MemoryProxy.js	186
7.107	APIIS_HOME/lib/Apiis/Extjs/src/data/JsonWriter.js	186
7.107	APIIS_HOME/lib/Apiis/Extjs/src/data/HttpProxy.js	186
7.107	APIIS_HOME/lib/Apiis/Extjs/src/data/Api.js	186
7.107	APIIS_HOME/lib/Apiis/Extjs/src/data/GroupingStore.js	186
7.107	APIIS_HOME/lib/Apiis/Extjs/src/data/Store.js	186
7.107	APIIS_HOME/lib/Apiis/Extjs/src/data/StoreMgr.js	186
7.107	APIIS_HOME/lib/Apiis/Extjs/src/util/History.js	186
7.107	APIIS_HOME/lib/Apiis/Extjs/src/util/KeyNav.js	186
7.107	APIIS_HOME/lib/Apiis/Extjs/src/util/UpdateManager.js	186
7.108	APIIS_HOME/lib/Apiis/Extjs/src/util/Format.js	186
7.108	APIIS_HOME/lib/Apiis/Extjs/src/util/core/TaskMgr.js	186
7.108	APIIS_HOME/lib/Apiis/Extjs/src/util/core/JSON.js	186
7.108	APIIS_HOME/lib/Apiis/Extjs/src/util/core/Observable.js	186
7.108	APIIS_HOME/lib/Apiis/Extjs/src/util/core/DelayedTask.js	186
7.108	APIIS_HOME/lib/Apiis/Extjs/src/util/ClickRepeater.js	186
7.108	APIIS_HOME/lib/Apiis/Extjs/src/util/CSS.js	186
7.108	APIIS_HOME/lib/Apiis/Extjs/src/util/Cookies.js	186
7.108	APIIS_HOME/lib/Apiis/Extjs/src/util/KeyMap.js	186
7.108	APIIS_HOME/lib/Apiis/Extjs/src/util/MixedCollection.js	186
7.109	APIIS_HOME/lib/Apiis/Extjs/src/util/Observable-more.js	186
7.109	APIIS_HOME/lib/Apiis/Extjs/src/util/XTemplate.js	186
7.109	APIIS_HOME/lib/Apiis/Extjs/src/util/TextMetrics.js	186
7.109	APIIS_HOME/lib/Apiis/Extjs/src/util/Date.js	186
7.109	APIIS_HOME/lib/Apiis/Extjs/src/dd/StatusProxy.js	186
7.109	APIIS_HOME/lib/Apiis/Extjs/src/dd/Registry.js	186
7.109	APIIS_HOME/lib/Apiis/Extjs/src/dd/DDCore.js	186
7.109	APIIS_HOME/lib/Apiis/Extjs/src/dd/DragTracker.js	186
7.109	APIIS_HOME/lib/Apiis/Extjs/src/dd/DropTarget.js	186
7.109	APIIS_HOME/lib/Apiis/Extjs/src/dd/ScrollManager.js	186
7.110	APIIS_HOME/lib/Apiis/Extjs/src/dd/DragZone.js	186
7.110	APIIS_HOME/lib/Apiis/Extjs/src/dd/DragSource.js	186
7.110	APIIS_HOME/lib/Apiis/Extjs/src/dd/DropZone.js	186
7.110	APIIS_HOME/lib/Apiis/Extjs/src/debug.js	186

7.1104	APIIS_HOME/lib/Apiis/Extjs/src/state/Provider.js	186
7.1105	APIIS_HOME/lib/Apiis/Extjs/src/state/CookieProvider.js	186
7.1106	APIIS_HOME/lib/Apiis/Extjs/src/state/StateManager.js	186
7.1107	APIIS_HOME/lib/Apiis/Load2.pm	186
7.1108	APIIS_HOME/lib/Apiis/GUI/Hash.pm	186
7.1109	APIIS_HOME/lib/Apiis/GUI/Excel.pm	186
7.1110	APIIS_HOME/lib/Apiis/GUI/XML.pm	186
7.1111	APIIS_HOME/lib/Apiis/GUI/FixPDF.pm	186
7.1112	APIIS_HOME/lib/Apiis/GUI/Ascii.pm	186
7.1113	APIIS_HOME/lib/Apiis/GUI/PDF.pm	186
7.1114	APIIS_HOME/lib/Apiis/GUI/HTML.pm	186
7.1115	APIIS_HOME/lib/Apiis/Load.pm	186
7.1116	APIIS_HOME/lib/Apiis/Init/Date.pm	186
7.1117	APIIS_HOME/lib/Apiis/Init/XMLForms.pm.org	186
7.1118	APIIS_HOME/lib/Apiis/Init/XML.pm	186
7.1119	APIIS_HOME/lib/Apiis/Init/Cache.pm	186
7.1120	APIIS_HOME/lib/Apiis/Init/Compat.pm	186
7.1121	APIIS_HOME/lib/Apiis/Form/new	186
7.1122	APIIS_HOME/lib/Apiis/Form/Init.pm	186
7.1123	APIIS_HOME/lib/Apiis/Form/form.dtd	186
7.1124	APIIS_HOME/lib/Apiis/Form/test.Form.ulf	186
7.1125	APIIS_HOME/lib/Apiis/Form/README.testing	186
7.1126	APIIS_HOME/lib/Apiis/Form/Event/Query.pm	186
7.1127	APIIS_HOME/lib/Apiis/Form/Event/Update.pm	186
7.1128	APIIS_HOME/lib/Apiis/Form/Event/printf_out.pm	186
7.1129	APIIS_HOME/lib/Apiis/Form/Event/Insert.pm	186
7.1130	APIIS_HOME/lib/Apiis/Form/Event/Clear.pm	186
7.1131	APIIS_HOME/lib/Apiis/Form/Event/CallForm.pm	186
7.1132	APIIS_HOME/lib/Apiis/Form/Event/Misc.pm	186
7.1133	APIIS_HOME/lib/Apiis/Form/Event/HandleDS.pm	186
7.1134	APIIS_HOME/lib/Apiis/Form/erster.png	186
7.1135	APIIS_HOME/lib/Apiis/Form/HTML/Submit.pm	186
7.1136	APIIS_HOME/lib/Apiis/Form/HTML/LabFrame.pm	186
7.1137	APIIS_HOME/lib/Apiis/Form/HTML/Extjs.pm	186
7.1138	APIIS_HOME/lib/Apiis/Form/HTML/CreateCSS.pm	186
7.1139	APIIS_HOME/lib/Apiis/Form/Tk/BrowseEntry.pm	186
7.1140	APIIS_HOME/lib/Apiis/Form/Tk/ScrollingList.pm	186
7.1141	APIIS_HOME/lib/Apiis/Form/Tk/Label.pm	186
7.1142	APIIS_HOME/lib/Apiis/Form/Tk/LabFrame.pm	186
7.1143	APIIS_HOME/lib/Apiis/Form/Tk/ButtonLib.pm	186
7.1144	APIIS_HOME/lib/Apiis/Form/Tk/DateEntry.pm	186
7.1145	APIIS_HOME/lib/Apiis/Form/Tk/Button.pm	186
7.1146	APIIS_HOME/lib/Apiis/Form/Tk/Message.pm	186
7.1147	APIIS_HOME/lib/Apiis/Form/Tk/TextField.pm	186
7.1148	APIIS_HOME/lib/Apiis/Form/Tk/Tabular.pm	186
7.1149	APIIS_HOME/lib/Apiis/Form/Tk/TextBlock.pm	186
7.1150	APIIS_HOME/lib/Apiis/Form/erster2.png	186
7.1151	APIIS_HOME/lib/Apiis/Form/WWWUtilities.pm	186
7.1152	APIIS_HOME/lib/Apiis/Form/Tk.pm	186
7.1153	APIIS_HOME/lib/Apiis/Form/Init/Config.pm	186

7.115	APIIS_HOME/lib/Apiis/Form/Init/Misc.pm	186
7.115	APIIS_HOME/lib/Apiis/Form/Formdevelopment.tex	186
7.115	APIIS_HOME/lib/Apiis/Form/forms/navigationbar_fields.xml	186
7.115	APIIS_HOME/lib/Apiis/Form/forms/RemoveBlanks.pl	186
7.115	APIIS_HOME/lib/Apiis/Form/forms/form.dtd	186
7.115	APIIS_HOME/lib/Apiis/Form/forms/test9-scrapie.frm	186
7.116	APIIS_HOME/lib/Apiis/Form/forms/test1.frm_	186
7.116	APIIS_HOME/lib/Apiis/Form/forms/Ablammung.mfrm	186
7.116	APIIS_HOME/lib/Apiis/Form/forms/q	186
7.116	APIIS_HOME/lib/Apiis/Form/HTML.pm	186
7.116	APIIS_HOME/lib/myWriter.pm	186
7.116	APIIS_HOME/lib/update/upd_core_01.pl	186
7.116	APIIS_HOME/lib/update/README.01	186
7.116	APIIS_HOME/lib/update/upd_transfer_01.txt	186
7.116	APIIS_HOME/lib/update/upd_AR_01.txt	186
7.116	APIIS_HOME/lib/ajax.js	186
7.117	APIIS_HOME/lib/popreport/dummy.dump	186
7.117	APIIS_HOME/lib/popreport/apiisrc	186
7.117	APIIS_HOME/lib/alt.js	186
7.117	APIIS_HOME/lib/hierArrays.js	186
7.117	APIIS_HOME/lib/form.js	186
7.117	APIIS_HOME/lib/formneu.js	186
7.117	APIIS_HOME/lib/general.js	186
7.117	APIIS_HOME/lib/apiis2javascript.js	186
7.117	APIIS_HOME/lib/menu_items.js.org	186
7.117	APIIS_HOME/lib/Popreport.pm	186
7.118	APIIS_HOME/lib/menu.js	186
7.118	APIIS_HOME/bin/model2xml.pl	186
7.118	APIIS_HOME/bin/agr-extract_files	186
7.118	APIIS_HOME/bin/add_gen_didier-32bit	186
7.118	APIIS_HOME/bin/is_valid_email	186
7.118	APIIS_HOME/bin/ped_completeness.pm	186
7.118	APIIS_HOME/bin/inbreeding_report	186
7.118	APIIS_HOME/bin/MGUI	186
7.118	APIIS_HOME/bin/upload.pl	186
7.118	APIIS_HOME/bin/hashtable	186
7.119	APIIS_HOME/bin/add_gen_didier_ulf	186
7.119	APIIS_HOME/bin/process_uploads.sh	186
7.119	APIIS_HOME/bin/Create_Population_tables.pl	186
7.119	APIIS_HOME/bin/file2inspool.pl	186
7.119	APIIS_HOME/bin/texfiles	186
7.119	APIIS_HOME/bin/log_of_Inbreeding.pm	186
7.119	APIIS_HOME/bin/READMEAdd_gen_didierReport.txt	186
7.119	APIIS_HOME/bin/LoadProject.sh	186
7.119	APIIS_HOME/bin/runexample.sh	186
7.119	APIIS_HOME/bin/add_gen_didier	186
7.120	APIIS_HOME/bin/cvstat	186
7.120	APIIS_HOME/bin/run_popreport_file	186
7.120	APIIS_HOME/bin/add_gen_didier-64bit	186
7.120	APIIS_HOME/bin/agr-run_parallel	186

7.120	APIIS_HOME/bin/mkdescr	186
7.120	APIIS_HOME/bin/run_popreport	186
7.120	APIIS_HOME/bin/show_rules	186
7.120	APIIS_HOME/bin/gnu_comp.txt	186
7.120	APIIS_HOME/bin/load_db_from_INSPool	186
7.120	APIIS_HOME/bin/additive.o	186
7.121	APIIS_HOME/bin/selection.pl	186
7.121	APIIS_HOME/bin/dbf2txt	186
7.121	APIIS_HOME/bin/WebForm.pl	186
7.121	APIIS_HOME/bin/PopulationReport.pl	186
7.121	APIIS_HOME/bin/load_stat	186
7.121	APIIS_HOME/bin/pedigree_loops.pl	186
7.121	APIIS_HOME/bin/xml2model.pl	186
7.121	APIIS_HOME/bin/mkxmlforms	186
7.121	APIIS_HOME/bin/.Rhistory	186
7.121	APIIS_HOME/bin/apiis-test-dependencies	186
7.122	APIIS_HOME/bin/InbreedingReport.pl	186
7.122	APIIS_HOME/bin/handle_pedi_file	186
7.122	APIIS_HOME/bin/pedicompl.pl	186
7.122	APIIS_HOME/bin/apiish	186
7.122	APIIS_HOME/bin/check_pedi_loop	186
7.122	APIIS_HOME/bin/cleanup_incoming.sh	186
7.122	APIIS_HOME/bin/READMEPopulationReport.txt	186
7.122	APIIS_HOME/bin/access_rights_ar_batch.pl	186
7.122	APIIS_HOME/bin/CreatePediStack.pl	186
7.122	APIIS_HOME/bin/mk_rand_string	186
7.123	APIIS_HOME/bin/cvs2cl	186
7.123	APIIS_HOME/bin/agr-extract_files.ulf	186
7.123	APIIS_HOME/bin/GUIN	186
7.123	APIIS_HOME/bin/test.LO	186
7.123	APIIS_HOME/bin/test.pl	186
7.123	APIIS_HOME/bin/datediff	186
7.123	APIIS_HOME/bin/extract_for_blupMEM.pl	186
7.123	APIIS_HOME/bin/Create_inbreeding_tables.pl	186
7.123	APIIS_HOME/bin/READMEInbreedingReport.txt	186
7.123	APIIS_HOME/bin/mkLOWform	186
7.124	APIIS_HOME/bin/add_gen_didier.f90	186
7.124	APIIS_HOME/bin/add_gen_didier.incl	186
7.124	APIIS_HOME/bin/ne_r_sloop_log_inbreeding.pl	186
7.124	APIIS_HOME/bin/Add_gen_didierReport.pl	186
7.124	APIIS_HOME/bin/test.Menu	186
7.124	APIIS_HOME/bin/test.JSON	186

Chapter 1

Introduction

In APIIS we define three groups of people working with the system. These are:

developers the guys (and may be gals) developing the routines usually hidden from those people implementing a system

implementors people who use the tools box to setup a running system, which pertains to a particular set of conditions. This may be the cattle population in one country

users these are the people who in the end use the system

Clearly, we may sometime find persons who are both implementors and users.

The developer's documentation covers all software that is hidden from the implementors. This means we are talking about the deep and sometimes dark vaults and sometimes even dungeons of APIIS.

You may be surprised by the section "Undocumented subroutines". Currently, the meta layer of APIIS is being rewritten to an object oriented mode. In this process, each routine that is handled will get its own documentation section in the Perl code. The subroutines listed here are of the non OO kind and will therefore have to be converted/replaced by the developers. Thus, this chapter serves more like a reminder for them to get its length towards zero.

Chapter 2

The APIIS Core

The APIIS core development system provides an object oriented interface to the basic structures.

Currently these parts build the core:

- Basic Initialization
- The Model File Interface
- The Database Interface

2.1 Basic Initialization

Each program in the APIIS suite has to start with the preinitialization:

```
#!/usr/bin/env perl
#####
# $Id: basic.tex,v 1.3 2005/03/08 11:02:34 heli Exp $
# Short description about this program
#####

BEGIN {
    use Env qw( APIIS_HOME );
    die "\n\tAPIIS_HOME is not set!\n\n" unless $APIIS_HOME;
    push @INC, "$APIIS_HOME/lib";
}

use strict;
use warnings;
use Apiis;
Apiis->initialize( VERSION => '$Revision: 1.3 $' );
```

The BEGIN block checks at compile time, if the environment variable \$APIIS_HOME is set. \$APIIS_HOME is the main starting point for the complete code base of APIIS. \$APIIS_HOME/lib is included into the library search path of Perl.

`use Apiis.pm;` loads the APIIS -system into your main namespace and provides the method `initialize()`. Usually you should pass the version of your program to `initialize`. When you commit this program to CVS, the cvs-tag `$Revision: 1.3 $` will be replaced by the current version and it will look like `$Revision: 1.3 $`. You can recall the version later with `$apiis->version`. If you don't use cvs then just hardcode the version: `Apiis->initialize(VERSION => '0.71');`

Every program which exceeds the intention of being a quick and dirty script for a small daily job should contain the following two lines before you start coding:

```
use strict;
use warnings;
```

You should not ask somebody else to help you track down some bugs in your code unless you used `strict` and `warnings`!

2.1.1 initialize

Besides performing some basic checks the main task of `initialize` is the creation of the global `$apiis`-structure, represented by the object reference `$apiis`.

2.1.2 Apiis::Init

The successful creation of the `$apiis` object implies some basic initializing before and adds various items:

- the base configuration from `$APIIS_HOME/apisrc`
- the I18N/L10N part
- methods to extend the `$apiis` structure modularly
- methods for logging to files or the system syslog
- the public interfaces to set an error status, check this status and document errors in detail
- handy methods to access the current day and time

The main setup for error handling is done in the package `Apiis::Errors`. On the main level, the `$apiis` object provides these public methods:

<code>\$apiis-></code>	<code>APIIS_HOME</code>	path to <code>\$APIIS_HOME</code>
	<code>APIIS_LOCAL</code>	path to <code>\$APIIS_LOCAL</code>
	<code>browser</code>	your favourite browser
	<code>check_status</code>	checks the error status
	<code>code_table</code>	usually table 'codes'
	<code>date_format</code>	the date format (US or EU)
	<code>entry_views</code>	hashref of entry-views
	<code>errors</code>	array(reference) of error objects
	<code>exists_database</code>	is the database joined to <code>\$apiis</code>
	<code>exists_form</code>	is a form joined to <code>\$apiis</code>
	<code>exists_model</code>	is the model file joined to <code>\$apiis</code>
	<code>fileselector</code>	choose a Tk fileselector
	<code>HOME</code>	path to <code>\$HOME</code>
	<code>join_database</code>	join the DataBase object into <code>\$apiis</code>
	<code>join_form</code>	join a named Form object into <code>\$apiis</code>
	<code>join_model</code>	join the Model object into <code>\$apiis</code>
	<code>language</code>	choose user language
	<code>localtime</code>	unformatted timestamp
	<code>now</code>	formatted timestamp
	<code>programname</code>	name of the invoking program
	<code>reserved_strings</code>	reserved strings for data
	<code>status</code>	error status
	<code>today</code>	formatted date
	<code>user</code>	current user
	<code>version</code>	version of the invoking program

See the POD-documentation for a detailed reference of these methods.

2.2 The Model File Interface

Usually you have to provide the model file, you want to use with your APIIS program. This is done with `$apiis->join_model('my_model_file');`. `join_model()` reads in the model file, parses it and creates public interfaces to all the information via an object, which is added to the `$apiis` structure via `Apiis::Init::_add_object`. The entry point for these methods is `$apiis->Model`.

An auxiliary module to find the model file in the `$APIIS_HOME` hierarchy is `Apiis::CheckFile`, which scans some likely directories for candidates. It also can be used for searching other configuration files like those for forms or reports.

The model file contains both some basic informations and – primarily – the complete database structure, including business rules.

The basic information can be accessed by these public methods:

<code>\$apiis->Model-></code>	
<code>basename</code>	basename of the model file
<code>check_level</code>	get/set check level
<code>db_driver</code>	database driver
<code>db_host</code>	database host machine
<code>db_name</code>	database name
<code>db_password</code>	database password of current user
<code>db_port</code>	database system port
<code>db_user</code>	database system (meta) user
<code>ext</code>	extension of model file name
<code>fullname</code>	full name of model file
<code>max_check_level</code>	maximum check level reached
<code>path</code>	path to model file
<code>table</code>	object for detailed table information
<code>tables</code>	array(ref) of all tablenames

For all tables in the model file, a table object is created which contains the information for this table. To get all columns of table 'animal' you could write:

```
my $table_obj = $apiis->Model->table('animal');
printf "Columns of table animal are: %s\n",
    join(', ', $table_obj->columns);
```

A more intuitive and shorter way is:

```
printf "Columns of table animal are: %s\n",
    join(', ', $apiis->Model->table('animal')->columns);
```

The methods of the table object are:

<code>\$apiis->Model->table(<name>)-></code>	
<code>check</code>	check rules of a column
<code>cols</code>	columns of this table
<code>columns</code>	columns of this table
<code>datatype</code>	meta-datatype of a column
<code>default</code>	default value of a column
<code>description</code>	description of a column
<code>foreignkey</code>	foreign key definition of a column
<code>index</code>	indices for this table
<code>indexes</code>	indices for this table
<code>indices</code>	indices for this table
<code>length</code>	default length of a column (forms)
<code>modify</code>	modify rules of a column
<code>name</code>	name of this table
<code>primarykey</code>	primary key definitions of this table
<code>rowid</code>	get/set rowid of this record
<code>sequence</code>	sequences defined for this table
<code>sequences</code>	sequences defined for this table
<code>triggers</code>	trigger definitions for this table

Most of the mentioned methods will be used rarely as they are implemented in the newer record object in a more consistent way. Some of them are even outdated, like the read/write method

```
$apiis->Model->table( $mytable )->rowid( $thisrowid );
```

A future rewrite of the model file interface will have a syntax similar to the record objects. Instead of

```
$apiis->Model->table($thistable)->foreignkey->($thiscolumn);
```

it likely will look like

```
$apiis->Model->table($thistable)->column($thiscolumn)->foreignkey;
```

This will give a consistent syntax according to this record object example:

```
$record_obj->column($thiscolumn)->foreignkey;
```

2.3 The Database Interface

When the Model object is joined into the `$apiis` structure, by default the database connection will be configured and initialized. Another object called `DataBase` is integrated into `$apiis`.

For rare situations you can circumvent this immediate connection to the database (e.g. if you want to drop the whole database during initialization) with a special parameter to `join_model()`. See the POD-documentation of `join_model()` for details.

The database connections are based on the configuration parts of the model file and the database specifics, found in `$APIIS_HOME/lib/<db_driver>.conf`.

If no errors occur, a `Apiis::DataBase::Init`-object will be created and joined into the `$apiis` structure as `$apiis->DataBase`.

Two connections to the database will be opened, one for the system user with full access to the database and the other as the currently connected user with access to the suite of views that reflect his authorization status. These two connections build the base for the public methods `sys_sql()` and `user_sql()`.

The following public methods are defined with `$apiis->DataBase`:

<code>\$apiis->DataBase-></code>	
<code>bindtypes</code>	database bindtypes
<code>commit</code>	commit changes to the database
<code>connect</code>	DB-specific connect string for DBI
<code>connected</code>	flag if DB is already connected
<code>datatypes</code>	datatypes of this DB for our metatypes
<code>dateorder</code>	default (?) date order of this DB
<code>datesep</code>	date separator
<code>db_has_sequence</code>	flag, if this DB has sequences
<code>disconnect</code>	method to disconnect from DB
<code>explain</code>	explain syntax (curr. not used)
<code>rollback</code>	rollback all changes
<code>rowid</code>	name of the rowid
<code>seq_next_val</code>	method that returns the next sequence value
<code>sequence_call</code>	how to call a sequence
<code>sys_dbh</code>	database handle for system user
<code>user_dbh</code>	database handle for common user

2.4 The Database Record Object

Still unwritten. ☹

But some POD already exists. ☺

At least here is the list of the public methods of `Apiis::DataBase::Record`:

<code>\$record_obj-></code>	
<code>action</code>	record action (like insert, update)
<code>addcolumn</code>	add a column to the record object
<code>check_level</code>	get/set check level
<code>column</code>	reference to a column object
<code>columns</code>	names of all columns
<code>decode_column</code>	decode on column level
<code>decoded</code>	decoded flag
<code>decode_record</code>	decode whole record
<code>delcolumn</code>	delete a column from the record object
<code>delete</code>	SQL delete action
<code>encode_column</code>	encode on column level
<code>encoded</code>	encoded flag
<code>encode_record</code>	encode whole record
<code>expect_columns</code>	which columns are expected by a fetch
<code>expect_rows</code>	how many records are expected by a fetch
<code>fetch</code>	SQL fetch/query action
<code>fk_table</code>	foreign key table
<code>indexes</code>	indexes of this table
<code>insert</code>	SQL insert action
<code>max_check_level</code>	max check level in model file
<code>name</code>	(table)name of this record
<code>new</code>	create new record object
<code>print</code>	print record (debug)
<code>rows</code>	number of rows returned by SQL action
<code>sequences</code>	sequences defined in this table
<code>tablename</code>	name of this table
<code>type</code>	type of this record object (database)
<code>update</code>	SQL update action
<code>value</code>	value returned by last SQL action
<code>values</code>	values returned by last SQL action

2.4.1 The Column Object

Besides record level methods, the record object contains one column object for each column of this table. The public methods to access the columns are:

<code>\$record_obj->column(<columnname>-></code>	
<code> check</code>	check rules of this column
<code> datatype</code>	datatype
<code> db_column</code>	database column name
<code> default</code>	default value from model file
<code> description</code>	description for this column
<code> extdata</code>	external data (array!)
<code> foreignkey</code>	foreign key definition of this column
<code> intdata</code>	internal data (scalar)
<code> length</code>	default length of this column (forms)
<code> modify</code>	modify rules of this column
<code> name</code>	name of this column (usually =db_column)
<code> updated</code>	updated flag

Example:

```
my ($fk_table, $fk_column, @rest) =
  $record_obj->column( $thiscolumn )->foreignkey;
```


Chapter 3

The Meta-layer

3.1 Data-flow schema

In Apiis both ordinary SQL statements and PseudoSQL statements are allowed. From the standard SQL92 syntax only `simpleinsert`, `update` and `delete` can be used. These statements get parsed and data are filled in the internal part of record object where the access rights check against the user rights take place. After successfully passing the access rights check the statement is executed via the meta-user account in the database - just like the ordinary PseudoSQL statements. The ordinary `Select` statements can be very complicated and hard to parse therefore they follow different route - these statements are not parsed, but directly executed via user account against the views in his own schema. These views are subset of the standard tables restricted with the user access rights, and have the same name as the original table. There the only item in the database that user has access to. The PseudoSQL statements get parsed, loaded in the external part of record object and then encoded, checked against access rights and executed via the meta-user account. The flow is shown on 3.1

3.2 PseudoSQL language definition

In this chapter the formal definition of the PseudoSQL language is described.

The PseudoSQL has only `INSERT`, `UPDATE`, `DELETE` and `SELECT` statement.

```
<pseudosql_statement> ::= <insert_statement> |  
    <update_statement> |  
    <delete_statement> |  
    <select_statement>
```

3.2.1 INSERT

```
<insert_statement> ::= 'INSERT INTO <table_name>
```

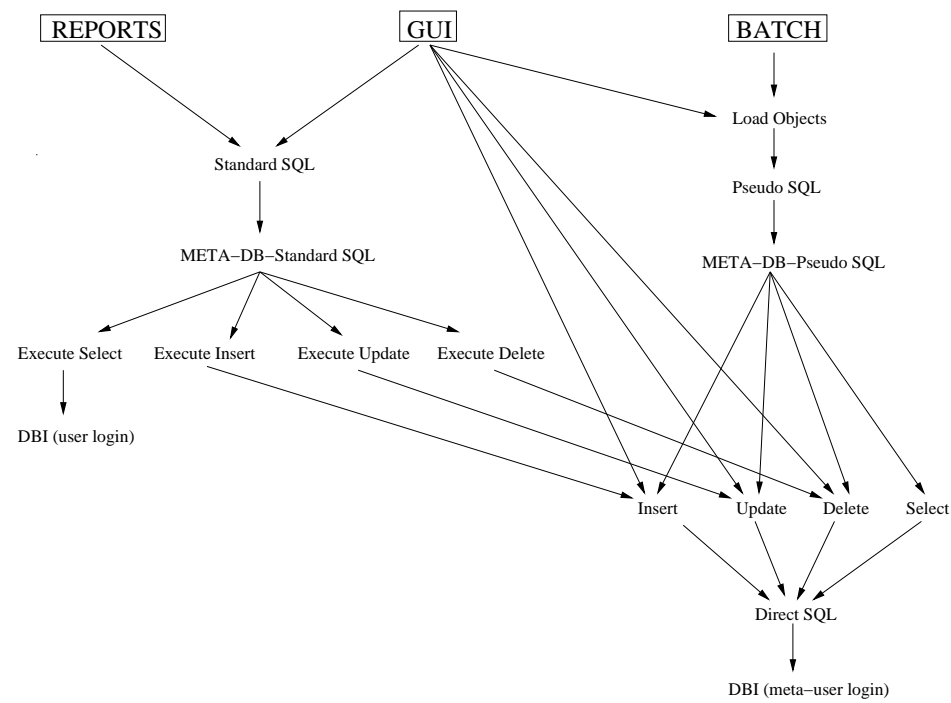


Figure 3.1:

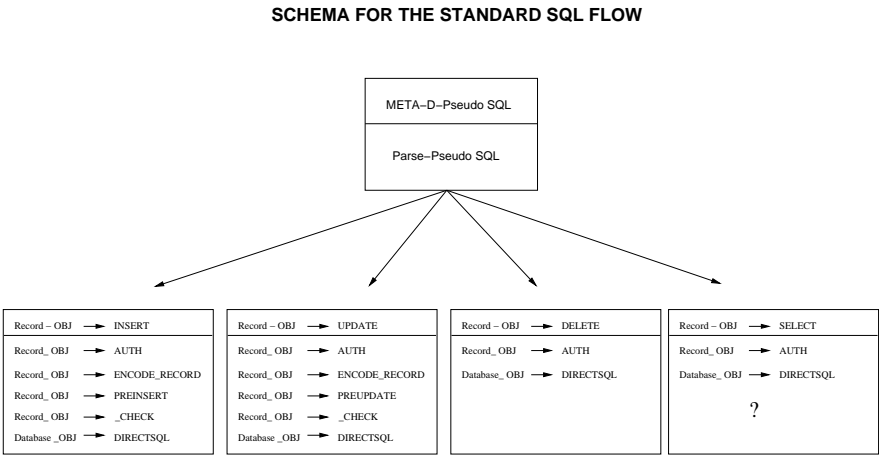


Figure 3.2:

```

    ( <column_name> {, <column_name> })
        VALUES ( <value> {, <value> } )'

<table_name> ::= <identifier>

<identifier> ::= LETTER{LETTER|DIGIT}

<column_name> ::= <identifier>

<value> ::= <scalar>|<string>|<variable>|<function>

<scalar> ::= DIGIT{DIGIT}|DIGIT{DIGIT}.DIGIT{DIGIT}

<string> ::= ''{LETTER|DIGIT}''

<variable> ::= ${hash_key}[<external_fields>]

<hash_key> ::= <identifier>

<external_fields> ::= [<field_list>]|[]
here [] means square brackets, not optional element

<field_list> ::= ''{<hash_key>{,<hash_key>}}''

<function> ::= concat( <item> , <item> {,<item>})

<item> ::= <value>|<variable>|<string>.<variable>|<variable>.<string>

```

3.2.2 UPDATE

```

<update_statement> ::= 'UPDATE <table_name>
                        SET <column_name> = <value> {, <column\_name> = <value> }
                        [WHERE <where_clause> ]'

<where_clause> ::= <column_name><operator><value>[ <AND|OR> <column_name><operator><value> ]

```

3.2.3 DELETE

```

<delete_statement> ::= 'DELETE FROM <table_name> [WHERE <where_clause> ]'

```

3.2.4 SELECT

```

<select_statement> ::= 'SELECT <variable> {, <variable> }
                        FROM <table_name>

```

[WHERE <where_clause>],'

3.3 PseudoSQL flow

Subroutine name: meta_db_PseudoSQL

Input: PseudoSQL - string

Output:

Description:

1. Parses the PseudoSQL string using the Parse_PseudoSQL subroutine - to be written
2. Executes the statement by call to a responding record object method: "insert", "update" or "delete";

Subroutine name: Parse_PseudoSQL

Input: PseudoSQL - string

Output: Record object filled with the data from the PseudoSQL string

Description:

1. Parses the input string - gets the action name, table, column names, column values and the where clause of the PseudoSQL - to be written
2. Creates links between LO keys and database columns. These links will be used for targeting the errors, that may come from the underlying levels - to be written
3. Modifies and encodes the where clause and query the database for the rowid of the first record responding to this where clause - to be written
4. Creates record(table) object for the table parsed from the string: my \$thisrecord = Apiis::DataBase::Record-> new(tablename => \$table,);
5. Fills the columns values from the PseudoSQL statement and the rowid in the responding column objects using the record object method "extdata" : \$record-> column(\$column_name)-> extdata(\$column_value);

3.4 Standard SQL flow

Subroutine name: meta_db_StandardSQL

Input: ordinary SQL - string

Output:

Description:

1. Recognizes the sql action and if it is "SELECT" executes it through the view system using subroutine "ExecuteSelect"
2. In case of INSERT, UPDATE or DELETE parses the SQL string using the statement object from APIIS/DataBase/SQL/Statement.pm

3. Calls the responding subroutine: “ExecuteInsert”, “ExecuteUpdate” or “ExecuteDelete”

Subroutine name: ExecuteSelect

Input: ordinary SQL SELECT statement- string

Output: Statement handle

Description:

1. Connects to the database with the user login and executes the Select statement. For each user we will have also a database account and user’s own schema(namespace) in the database. This schema will have the same name as the user database login and in this schema all original tables will be masked with views with the same names. For each table we will have a view that is a subset of the original table based on the access rights of the user. The user will have only read access to his schema and no access rights for the rest of the database - to be written

Subroutine name: ExecuteInsert

Input: parsed sql statement object

Output:

Description:

1. Creates record(table) object for the table returned by the statement object;
2. Fills the columns values from the statement in the responding column objects using the record object method “intdata”
3. Executes the statement by call to a responding record object method: “insert”;

Subroutine name: ExecuteUpdate

Input: parsed sql statement object

Output:

Description:

1. Creates record(table) object for the table returned by the statement object;
2. Fills the columns values from the statement in the responding column objects using the record object method “intdata”
3. Queries the database with the where clause returned from the statement object and gets the oids of all responding records;
4. Looping through the query results, fills the oid column in the record object with the value from the query and executes the statement by call to a responding record object method: “update”;

Subroutine name: ExecuteDelete

Input: parsed sql statement object

Output:

Description:

1. Creates record(table) object for the table returned by the statement object;
2. Queries the database with the where clause returned from the statement object and gets the oids of all responding records;
3. Looping through the query results, fills the oid column in the record object with the value from the query and executes the statement by call to a responding record object method:“delete”;

3.5 Used record object methods

Method name: insert

Input: Record object filled with the data from the PseudoSQL string

Output:

Description:

1. Verifies if the user has appropriate rights to access the filled columns in the record object using the record object method “auth”. Normally at this stage the record object should contain only user data and the meta-fields are not set by the system yet;
2. If the record object is not encoded applies the modify rules from the model file to the external record data using record object method “_modify”;
3. If the record object is not encoded, encodes the external values to internal ones using the method “encode_record”;
4. Runs all "PREINSERT" triggers - this is the place where the meta-fields like last_change_user are filled;
5. Checks the business rules using the record object method “_check”. The checking is done always on the internal values;
6. Creates ordinary SQL statement and executes it via “directsql” method of the database object

Method name: update

Input: Record object filled with the data from the PseudoSQL string

Output:

Description:

1. Verifies if the user has appropriate rights to access the filled columns in the record object using the record object method “auth”. Normally at this stage the record object should contain only user data and the meta-fields are not set by the system yet;
2. If the record object is not encoded applies the modify rules from the model file to the external record data using record object method “_modify”;
3. If the record object is not encoded, encodes the external values to internal ones using the method “encode_record”;

4. Runs all "PREUPDATE" triggers - this is the place where the meta-fields like last_change_user are filled;
5. Checks the business rules using the record object method "_check". The checking is done always on the internal values;
6. Creates ordinary SQL statement and executes it via "directsql" method of the database object

Method name: delete

Input: Record object filled with the data from the PseudoSQL string

Output:

Description:

1. Verifies if the user has appropriate rights to access the filled columns in the record object using the record object method "auth". Normally at this stage the record object should contain only user data and the meta-fields are not set by the system yet;
2. Runs all "PREDELETE" triggers - in case of delete there are not any meta-fields to be filled, but maybe we will need this trigger for another purpose - referential integrity check;
3. Creates ordinary SQL statement and executes it via "directsql" method of the database object

Method name: auth

Input: Record object filled with the data from the PseudoSQL string

Output: Additional where clause for the sql statement - string

Description:

1. Reads from the database the user access rights for this table and action
2. Verifies if the user has appropriate access to these columns
3. Generates additional where clause to filter the owner

Chapter 4

The Access Control

4.1 Introduction

The main tasks of the security system will be to ensure proper access for the users and protect the structure and the content of any APIIS databases from the unauthorised actions.

APIIS needs guarantees that the persons connecting to the system are really the ones they claim to be and also has to control the actions each person is trying to perform.

4.2 Requirements for the access control system

The requirements for the APIIS security system are split on three parts:

- general requirements
- software requirements
- database requirements

4.2.1 General requirements

- access to the APIIS System is controlled by the login and the password
- access rights control is based on the RBAC¹(shortly saying the access rights are write down as a policies and the policies are assign to the roles)
- access rights are granted to users through the role groups
- role group consist of the roles or other role groups
- assigning users to the groups, group to the other groups or roles to the groups is controlled by the special constraints which are defined by the administrator (to prevent a situation where two excluding definitions are set together)

¹Role Based Access Control [2],[3]

4.2.2 Software requirements

- all APIIS software is placed in the secure space on the server - administrator account
- developers have full access to the APIIS file system
- normal users have access to read for the the APIIS file system
- all applications which cause modifications in the database (f.e. batch jobs) are restricted by the access rights
- accessing the graphical interface, form tools, report tools is verified by defined access rights

4.2.3 Database requirements

- database is created and controlled by the administrator
- users have not direct access into the database
- access rights to the database granted by the administrator (for all users)
- creating new database and new user is allowed only for the administrator
- actions on the database objects like creating, altering, dropping are revoked from the users
- access rights are specified for each table, column and the content of the column (record)
- operations on data like insert, update, delete and select are verified by the user access rights
- direct operations on data like insert, update, delete are not allowed for the user
- all modifications on data carried through the administrator
- selecting data permitted for the user through the views created by the administrator

4.3 The basic foundations - setting APIIS software

4.3.1 System architecture

There are three logical machines in an APIIS database setup:

- client machine - the machine at which users operate
- APIIS server - the machine that all users connect to
- Database server - the machine that runs the backend database

Clearly, all three logical machines can reside on one or more physical computers. The client machine is the computer from where user fire up the web browser or just connect to the APIIS Server via SSH protocol².

The APIIS Server takes care of the authentication of the users, connections to the database server and presenting data back to the users. Thus, to work directly on the APIIS server (via SSH), user must have an account on operating system level (see section 4.4.1) and also account on the APIIS system level (see section 4.4.2). The work with the web browser required only this second type of account.

The database server is the place where the database is stored. The connections into the database are supported by the APIIS Server. The connections to the database are distinguished on two groups: modifications on data and reading data. The modifications are executed only through the `meta_user` connection (see 4.3.3). Reading data is handled by the direct user connection.

4.3.2 APIIS core

APIIS core software resides on the APIIS Server and is used by the all users. This means that programs are executed on the server and all users use the same libraries and modules.

Definition of security for the APIIS software is based on the usage some secure space on the server and also on the operating system features³. Secure space is received by creating a special administrator account (OS account). Administrator is the owner of the files and this means that he has full rights to reading, writing and executing. Access to these files for the other users is handled by the special groups (see description of Linux groups). Each file has defined group to which belong and a special rights for this group (reading, writing and executing). In this case all files are defined in the administrator group (group is created with the administrator user account). This group can be ascribed to each operating system user and with this group user should have rights only to reading. If some of the files have to be fully restricted for the users (some administrator modules) than this can be done by removing all rights from the group and from the other users. In such case only administrator has access to these files.

There is also possibility to install separate copy of APIIS software for each user. In such case users must have operating system account because the software is installed in his home directory.

4.3.3 APIIS projects

All projects based on the APIIS core have separate databases. In each of these database we have to create the `meta_user` account. Then the `PUBLIC` schema has to be removed and the database must be created in the `meta_user` schema⁴ which is created with the `meta_user` account. In result the `meta_user` has full access right to the database and only he can make direct modifications on the

²Secure Shell [4]

³The setup described here should be fullfilled during software installation

⁴Schema is essentially a namespace which contains named objects like tables, views whose names duplicates those of other objects existing in other schema's (PostgreSQL [1]).

database content. The important thing is that the meta_user name have to be exactly the same like the name of the user defined in the model_file of particular project. There is only one model_file for all users registered in the project.

The general schema how the system is implemented is shown on Figure 4.1.

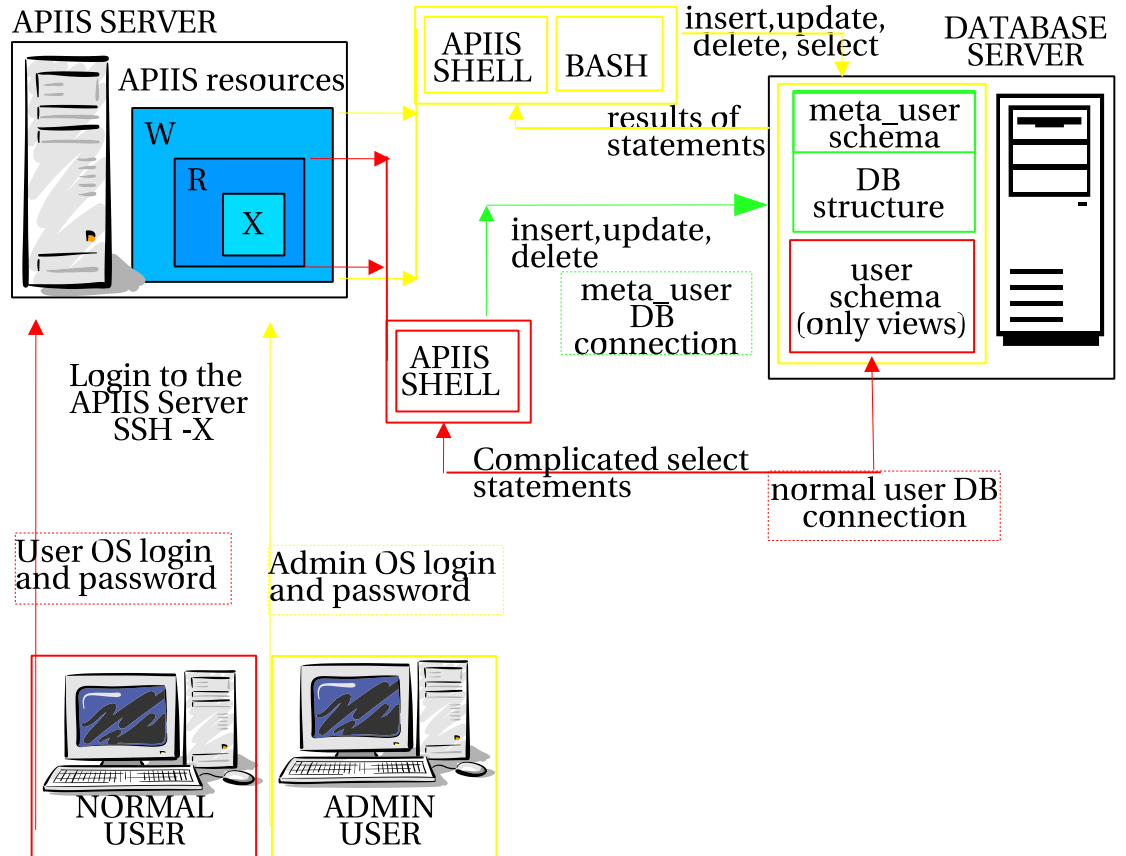


Figure 4.1: General schema for the access control system

4.4 Defining users

Only registered users can work in the system. Each user can be registered in the system on the two levels:

- operating system level (not required for the each user),
- APIIS system level (required for the each user).

4.4.1 Registering user on the operating system level

This type of account is needed to work directly on the APIIS Server and it is related only to this users, which use the APIIS Shell or run some batch jobs. In such case user must have an account (login and password) in the operating system of APIIS Server.

Standard Linux user account can be created by executing the following command:

```
adduser [login] -g [login] -G [APIIS administrator group]
```

During creation of account, the user is also assigned to the required Linux group (see 4.3.2).

At the end user APIIS_HOME path have to be defined in .bash or .profile file. Path have to be redirected to the APIIS administrator space where the software is kept (see 4.3.2).

Example:

```
export APIIS_HOME=/home/apiis_administrator/devel/apiis
```

4.4.2 Registering user on the APIIS system level

General APIIS account which is created on this level is required to work with the APIIS System. On the basis of this account the access rights for the user are created and then checked. The data about user are stored in the database (see section 4.13, figure 4.5: AR_Users). The following information about each user is collected:

- login
- password
- db_unit - foreign key to the unit table where the personal information about user is stored
- country
- language
- marker - the information about the ownership of the data
- disabled - this column is used to the locking of the user account. The flag of this column is always checked during the logging process and it can be set as YES (user can not login to the system) or NO (user can login to the system). There is also possibility to lock more then one user in the same time. This can be done by the lock of the user group to which the users are assigned
- status, last_login, last_activ_time - these three columns are used to controlling the user login time and to checking the user current status (see section 4.10.1).

As an example of the user data, you can see Table 4.1.

user_id	login	password	db_unit	country	language	marker
1	kloss	*****	22	German	DE	
2	jkowal	*****	455	Poland	Polish	PL

disables	status	last_login	last_activ_time
NO	ACTIVE	2005-07-29 09:38:28	2005-07-29 11:12:45
YES	INACTIVE	2005-05-12 11:12:45	2005-05-12 12:12:45

Table 4.1: Users table

The APIIS system is based on the PostgreSQL database and to work with it user needs also database account. PostgreSQL account is created automatically during the creation of APIIS account. The login and the password are exactly the same like these defined for the APIIS account. This database account is needed for log-in to the system and also to give user the possibility of executing SQL SELECTs. These SELECT statements are executed on the views⁵ which are created in the user schema on the basis of user access rights. The actions like insert, update, delete are effected by the meta_user (4.3.3). The meta_user is responsible for all modifications in the database and nobody else can do this. When the user executes a DML, the connection to the database is established from the meta_user. Real user name is used to check user access rights. Then the meta_user run all processes if the user has authorisation for this action. Real user name is sent as a normal data for the meta fields (last_change_user). All other actions like creating, dropping and altering some objects are revoked from the user (even after log-in in to database from the command line). The user can not also create new users and databases.

4.5 Granting access rights to the user

The access rights are granted to the user by the role groups. Each registered user should be assigned at least to the one group (f.e. own which is created during the registration process). The information about groups assigned to the users is stored in the database table (see section 4.13, figure 4.5: AR_User_Groups):

user_id	group_id
1	1
1	3

Table 4.2: Relations beetwen users and groups

group_id is a foreign key to the group's table where the group definitions are stored (see section 4.13, figure 4.5: AR_Groups table).

The user is allocated for the group by the administrator. If the administrator wants to add the user to the group, first he has to check that the user

⁵The view is, in essence, a virtual table. It does not physically exist. Rather, it is created by a query joining one or more tables.

group_id	group_name	group_type	group_content	group_desc
1	system_task_administrator	st_group	Roles	description
2	database_administrator	dbt_group	Roles	description
3	breeder	dbt_group	Groups	description

Table 4.3: Groups

can be really assigned to this group - checking that the group can cooperate with the groups which are currently defined for this user. This process is done automatically on the basis of the group constraints. The group constraints qualify which groups can not be used in the same time by the one user. They are stored in the separate table in the database (see section 4.13, figure 4.8: AR_Group_Constraints).

group_cons_id	group1_id	group2_id	group_cons_type
1	2	3	user-group-cons
2	6	4	user-group-cons

Table 4.4: Group constraints

The fields group1_id and group2_id in the table are foreign keys to the table groups (Table 4.3). The algorithm, which verifies the groups, takes from the user the current list of his groups (from table 4.2). The values from the list are set together one by one with the id of the new group which we want to add. Each couple of values is used as a condition for the WHERE clause in the following SQL statement:

```
SELECT group_cons_id FROM ar_group_constraints WHERE
  ((group1_cons_id='user_defined_group' and
group2_cons_id='new_group') or (group1_cons_id='new_group' and
group2_cons_id='user_defined_group')) and
  (group_cons_type='user-group-cons')
```

For each couple of groups one SELECT is executed. When all combination of groups are positively verified (no results for each combination) then the user can be appraised to the group. If there is a result then this means that the constraints are defined for this combination and the new role group can not be added to the current set of groups defined for the user. The algorithm is not stooped in this point and it just go through the all combinations. All results are collected and then they are showed to the administrator. The administrator has clear picture which groups are in the conflict with the new group. The constraints for the groups are optional and it should be defined only if they are needful (the decision stay with the administrator). In the section 4.9 you can read how the constraints are defined.

4.6 Access rights for the system tasks

4.6.1 Definitions of the access rights

This access control definition is designed for the scripts, forms, reports, interface, subroutines and all other actions which are executed on the basis of APIIS software (I called these action as a system tasks). The administrator of the system has to be sure that the user runs only these tasks which are allowed for him. This means that every user has to have defined access rights for the each system task. The definition of the access rights is based on the roles - roles based system (RBAC⁶). In this type of system each role is a definition of the group of the access rights. In the roles, the access rights are defined via policies. In our case each policy defines access to one system task. All roles are grouped and they are assigned to the user groups. The whole structure of access control for the system tasks is defined in the following manner: the policies are ascribed to the one or more roles, the roles are ascribed to the one or more role groups, the role groups are ascribed to the one or more user or to the next role groups. The information about access rights needed to control system tasks is stored in the three following tables (see section 4.13, figure 4.7):

- roles table (AR_Roles) - this table stores information about roles. The role definition is a set of role name and the role type where the role type can be defined as ST (System Task) or DBT (Database Task). In this case the role should be defined as a ST.

role_id	role_name	role_type
1	sys_admin_role	ST
2	public_role	ST
3	db_admin_role	DBT

Table 4.5: Roles table

⁶Role Based Access Control [2],[3]

- system task policies table (AR_StPolicies) - this table stores information about system tasks. Each system task consist of the name and the category. The category of the system task can be defined as: program, www, form, report, action.

stpolicy_id	stpolicy_name	stpolicy_type
1	runall_ar.pl	program
2	enter data	www
3	add new user	action
4	Number of animals in year 2004	report

Table 4.6: Policies for the system tasks

- link table (AR_Role_StPolicies) - it joins roles with the policies together.

4.6.2 Checking of the access rights - logging to the system

There are two ways to work with the APIIS system:

- directly on the APIIS server (APIIS Shell, batch jobs)
- through the web browser (WWW service).

If user wants to work directly on the APIIS server first he has to connect via ssh to the server (OS login and password - see section 4.4.1). After log-in to the server, special APIIS Shell is activated for the user. In the APIIS Shell user has to choose a project name (to which he want to login) and enters his APIIS login and the password (see section 4.4.2). If the data are consistent then the meta_user has to check the user access rights for the system task. The meta_user log-in to the database (the internal system connection) and checks which tasks user can execute. The result of this checking are returned as a list of the allowed jobs. This list is loaded in to the APIIS Shell. Finally user has only these actions in the Shell to which he is in the right. The symbolic schema is shown on Figure 4.2.

There are users which can have possibility to run batch jobs from the command line. In such case after log-in to the APIIS Server via ssh APIIS Shell is not activated automatically. The procedure of checking access rights for the programs which are run directly from the command line is exactly the same like during log-in via APIIS Shell. Difference here is that the meta_user checks access right only for the currently executed task.

Second variant to work with the APIIS is the web browser. Here instead of connection to the server user has to specify in his web browser the correct address to the APIIS internet page. Login procedure is exactly the same like for the APIIS Shell.

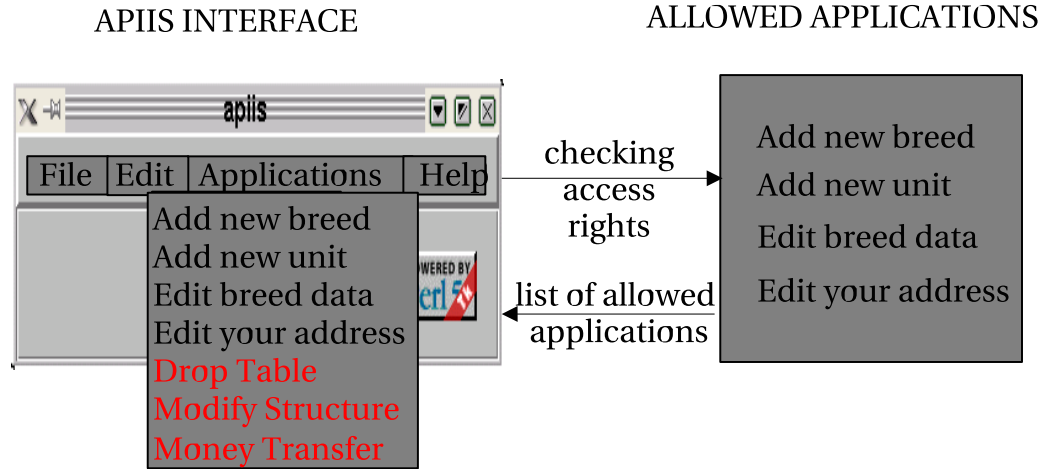


Figure 4.2: Logging to the APIIS system (red fields are not allowed)

4.7 Access rights for the database and the content of the database

This access control definition is designed for all action related to the database (database tasks). We define two different methods for checking access rights on the database level. The choice of which, depends on type of the SQL Statement. One applies only to the *insert*, *update* and *delete* statements while another is used for the *select* statements. The type of the SQL Statement is recognised on the beginning and then the relevant method is launched.

4.7.1 Method for the insert, update and delete statements

This route is especially for the insert, update, delete statements and can be described by the following steps:

1. All actions go through the meta_layer where the action is checked for the user
2. The arguments in the action (table/columns) are matched with the access rights defined for this user,
3. If the user doesn't have required access rights for the table/columns set the action is aborted, else access the record is checked in the next point.
4. Algorithm checks if the statements will be executed on the set of data which are allowed for the user. If DML touches the record which is out of user area then the action is cancelled.

4.7.1.1 Definition of the access rights

Definition of the access rights for the database tasks is based on the same structure like the definition of system tasks (4.6.1). In this case access rights are also ordered by the roles where the role defines access rights to the group of tables, columns, records. The roles are assigned to the role groups. Each role consists of one or more policy. The policy consists of action (INSERT/UPDATE/DELETE), table name, column names for this table and the descriptor, where the descriptor specifies the sets of data on which user can operate. The information about access rights for the database tasks is also stored in the database (see section 4.13, figure 4.6). In this case the same roles table which was defined for the system tasks is used (Table 4.5). The difference is only in the role type definition because here the role is defined as a DBT (Database Task). Besides, there are three additional tables and the view:

- tables (AR_DbtTables) - keeps information about the tables defined in the modelfile and their columns.

table_id	table_name	table_columns
1	breeds	breed_id country_id lean_meat_avg
2	breeds	breed_id tax_id mcname
3	breeds	breed_id lang_id intname
4	animal	db_animal birth_dt db_sex name
5	breeds	breed_id mcname country_id tax_id
6	breeds	breed_id mcname
7	breeds	breed_id mcname tax_id dailygain
8	breeds	-
9	animal	-

Table 4.7: Tables

- descriptor (AR_DbtDescriptors) - table holds the definitions of filters for the records. Descriptor can be defined as any column from the database. Each descriptor consist of the column name and the value for this column, where this last can be defined as a single value, list or range (range can be defined only for the numerical values). In case of list, there is a limitation related to the number of elements. If the list has more then 2000 elements then the special view in the user schema is created which will return values for this list. The name of this view is put in to the descriptor definition. If the descriptor is based on the foreign key than the internal representation of foreign key numbers is used as a values.

descriptor_id	descriptor_name	descriptor_value
1	lean_meat_avg	60-74
2	tax_id	5,6,7
3	owner	PL
4	db_animal	1-50
5	db_sex	72
6	lean_meat_avg	60-74
7	tax_id	5,6,7
8	owner	DE
9	db_animal	1-10
10	db_sex	72
11	tax_id	1,2
12	carcassweight	300-400
13	owner	PL,DE,FR,IT, ...
13	owner	PL,DE
14	owner	FR
15	tax_id	3
16	dailygain	24-56
17	NOT tax_id	1,2,3

Table 4.8: Descriptors

- database task policies - this table stores policy definitions which are a joins of records from tables: descriptor, table and codes. Table codes stores the SQL action names (1-INSERT, 2-UPDATE, 3-DELETE, 4-SELECT). The important thing is that the descriptor has to be always specified as a column of table which is used in the policy definition.
- user access view - the view is created in the user schema and keeps individual access rights of the user. The name of the view is derived from the user name.

policy_id	action_id	table_id	descriptor_id
1	1	1	1
2	1	2	2
3	1	3	3
4	1	4	4
5	1	4	5
6	2	1	1
7	2	2	2
8	2	3	3
9	2	4	4
10	2	4	5
11	3	8	2
12	3	9	4
13	4	5	11
14	4	5	12
15	4	5	13
16	4	6	14
17	4	6	15
18	4	7	16
19	4	7	17
20	4	8	4
21	4	8	5

Table 4.9: Database task policies table

4.7.1.2 Checking of the access rights

The procedure of checking access rights is executed for each SQL statement separately. Each SQL statement (from LO, forms, interface or other program), excluding SELECT, is parsed and the results are put into the special structure (record object). The information about SQL statement needed for the checking of access rights is taken from this structure.

4.7.1.2.1 Checking insert statement

1. Getting the action name, table name and the column names from the SQL statement which user want to execute. This information is taken from the record object.
2. Verifying user access rights for the action and the table.
Special "SELECT" statement is executed on the user access rights view. The action name and the table name (received in step 1) are used as arguments in the WHERE clause. It returns allowed column names and descriptors for defined table and action.
If there is some result from the SELECT statement then the access rights are valid for the action and the table and we can go to step 3. If there is no result (no record returned) user is not allowed to execute his SQL query and the algorithm is stopped.

action	tablename	columnnames	descriptor_name	descriptor_value
insert	breeds	breed_id country_id lean_meat_avg	lean_meat_avg	60-74
insert	breeds	breed_id tax_id mcname	tax_id	5,6,7
insert	breeds	breed_id lang_id intname	owner	PL
insert	animal	db_animal birth_dt db_sex name	db_animal	1-10
insert	animal	db_animal birth_dt db_sex name	db_sex	72
update	breeds	breed_id country_id lean_meat_avg	lean_meat_avg	60-74
update	breeds	breed_id tax_id mcname	tax_id	5,6,7
update	breeds	breed_id lang_id intname	owner	PL
update	animal	db_animal birth_dt db_sex name	db_animal	1-10
update	animal	db_animal birth_dt db_sex name	db_sex	72
delete	breeds	-	tax_id	5,6,7
delete	breeds	-	db_animal	1-50
select	breeds	breed_id mcname country_id tax_id	tax_id	1,2
select	breeds	breed_id mcname country_id tax_id	carcassweight	300-400
select	breeds	breed_id mcname country_id tax_id	owner	PL,DE
select	breeds	breed_id mcname	owner	FR
select	breeds	breed_id mcname	tax_id	3
select	breeds	breed_id mcname tax_id dailygain	dailygain	24-56
select	breeds	breed_id mcname tax_id dailygain	NOT tax_id	1,2,3
select	animal	db_animal birth_dt db_sex name	db_animal	1-50
select	animal	db_animal birth_dt db_sex name	db_sex	72

Table 4.10: User access rights view

3. Verifying user access rights for the columns.

Set of column from user SQL is matched with the sets of columns which are defined in the policies. If the algorithm finds the definition which is identical (or if the policy definition contain all column from user SQL) then the descriptor of this policy is collected (the order of column can be different but the names have to be the same). Algorithm goes through the all records returned in step 2 and accumulates all descriptors. Error message (no access rights) is generated in case if there is no applicable column definitions in the user rights.

4. Verifying user access rights for the record .

Now we have to prove all descriptors returned in the previous step. The value of each descriptor is set together with the value of the corresponding column from the user SQL⁷. If the value from user SQL is in the right with the descriptor value then the next pair of value is checked. If there is no compatibility for some pair of value then the error message is printed and action is stopped. The process of access rights checking is finished successfully if data introduced by the insert are contained in the user limitations.

Examples:

⁷if the descriptor value is defined as a list or range then the value from user SQL is searched on the list or it is collated with the range; in case of view the select statement is executed to verified this value.


```

(1) INSERT INTO breeds(breed_id,country_id,lean_meat_avg)
    VALUE (50000055,500000001,68);
(2) INSERT INTO breeds(breed_id,country_id,lean_meat_avg)
    VALUE (50000055,500000001,45);
(3) INSERT INTO breeds(breed_id,tax_id)
    VALUE (50000055,6);
(4) INSERT INTO breeds(breed_id,country_id,tax_id,lean_meat_avg)
    VALUE (50000055,500000001,7,45);
(5) INSERT INTO breeds(breed_id,lang_id,intname)
    VALUE (50000055,300000001,'name');

```

If we look at our view (Table 4.10) then: - the first insert can be executed by the user because the lean_meat_avg is 68 and allowed range is 60-74
 - the second insert can not be executed because lean_meat_avg is out of defined range
 - the third insert can be executed
 - the forth insert can not be executed because there is no such set of column definitions in any policy.
 - the fifth insert can be executed if the owner name which will be inserted to the record is defined as PL (the owner is a special case which is existing only in EFABIS project and it is taken from the user table).

4.7.1.2.2 Checking update statement The procedure of checking access rights for update is exactly the same like this defined for the insert. The differences are only in steps 2 and 4. In step 2, the parameter action for the WHERE clause is defined as UPDATE. In step 4 descriptors are compared with the values of record which will be updated by the user (in the INSERT they are compared with the values which are introduced by the user).

Examples:

```

(1) UPDATE breeds SET breed_id='50000045',mcname='new mcname'
    WHERE breed_id=444446;
(2) UPDATE animal SET birth_dt='2000-09-02', db_sex=73
    WHERE db_animal=444556;
(3) UPDATE animal SET birth_dt='2000-09-02', name='some name'
    WHERE db_animal>1 and db_animal<10 and db_sex=73;

```

In our examples the first update can be executed if the tax_id of existing record is defined as 5 or 6 or 7. The second update can not be executed because db_animal is out of the range. The third record can be also not executed because action is allowed only for the records where db_sex has 72 value (in this case db_animal is correct).

4.7.1.2.3 Checking delete statement In case of DELETE statement the algorithm works the same like for UPDATE with exclusion of step 3 (DELETE statement is executed on the whole record and the columns are not checked).

The symbolic schema of modifying database content is shown on the Figure 4.3.

All modifying query (insert,update,delete) are managed by the metauser (see ??).

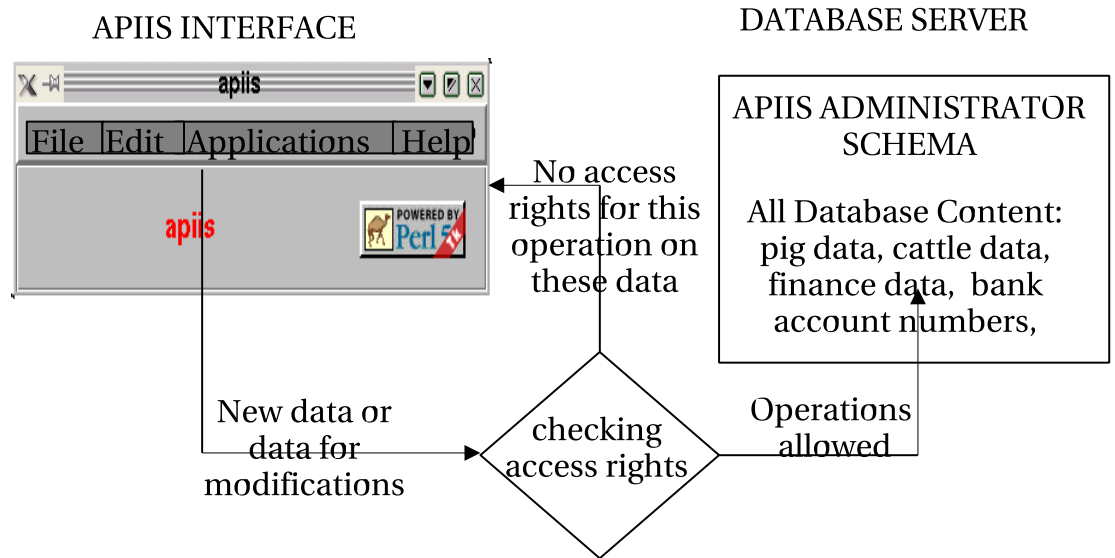


Figure 4.3: Modifying the database content

4.7.2 Method for the public select statements

Access rights for the public SELECT statements follow a different route from the method which was described in the previous section. This route is different, because the parsing of a complex SELECT statement (placing elements of the SQL query into the record object) is too complicated. In this case SELECT statements are not handled by the meta_user but by the real user with his direct database connection. They are executed on the views located in the user schema. User can access only these views which are created in his schema. Each view contains only those rows and columns that the user is allowed to access (on the basis of his access rights).

4.7.2.1 Definition of the access rights:

Access rights for the selecting data are defined in the same way like were defined for *update*, *insert* and *delete* (see 4.7.1).

4.7.2.2 Creating views:

Each user view is created separately. Always for each table one view is created. At the beginning list of all allowed table names is taken from the user access rights view (only these table names on which user can execute SELECT statement).

Then the following steps have to be accomplished to create view for each table from the list:

1. Creating list of basic columns for the view.
The algorithm takes from the user access view all column names for the

action	tablename	columnnames	descriptor_name	descriptor_value
select	breeds	breed_id mcname country_id tax_id	tax_id	1,2
select	breeds	breed_id mcname country_id tax_id	carcassweight	300-400
select	breeds	breed_id mcname country_id tax_id	owner	PL,DE
select	breeds	breed_id mcname	owner	FR
select	breeds	breed_id mcname	tax_id	3
select	breeds	breed_id mcname tax_id dailygain	dailygain	24-56
select	breeds	breed_id mcname tax_id dailygain	NOT tax_id	1,2,3
select	animal	db_animal birth_dt db_sex name	db_animal	1-50
select	animal	db_animal birth_dt db_sex name	db_sex	72

Table 4.11: The same user access rights view

table which is currently treated. The column names are taken from the each policy definition and then they are merged together in to the one list (duplicates of columns are removed). This list is needed to create basic view structure.

2. Creating basic SQL statement needed to produce view.
This first part of the SQL statement is defined on the basis of the columns which we got in the previous step.

*CREATE VIEW user_schema.treated_table AS SELECT list of
basic columns FROM meta_user_schema.treated_table WHERE
oid=NULL*

The "where clause" is needed here to create empty view structure. Now we have to add the filtration for the columns and the records according to the descriptor definitions.

3. Defining filtering extensions for the basic SQL statement.
The records are filtered by the additional SELECT statements which have to be defined separately for the each unique set of columns. SELECTS are created one by one and for each of them the following actions are effected:
 - At first the column for the SELECT are prepared. Treated set of columns is compared to the basic list from step 1. If some column is missing in the treated set then NULL expresion is placed instead of column. The order of column for this query has to be exactly the same like the order of basic column.
 - When the columns are ready then the WHERE clause is fixed. Thus all descriptors assigned for considered collection of columns have to be included. Each of the descriptors is joined to the WHERE clause by the AND operator. If descriptor has more than one value defined then the one condition from these values is created. In this case value are link by the OR operator⁸ and then they are added to the WHERE clause. It can be also that the value of descriptor is related to the

⁸This is true only if we have a list of value, in case of the range values are merged by AND operator

view and then the information are taken by the additional subquery (construction: descriptor IN (SELECT)).

If descriptor name is defined with the NOT prefix, the NOT expression is added to the WHERE before this element⁹.

- The complete SELECT is added to the basic SQL statement (from step 2) by the UNION expression. After this the next set of columns is taken into the process.
- After last SELECT final SQL is executed and the view for the table is created.

Example:

```
CREATE VIEW user\_schema.breeds as
  SELECT breed\_id, mcname, country\_id, tax\_id, dailygain FROM breeds
UNION
  SELECT breed\_id, mcname, country\_id, tax\_id, NULL FROM breeds
  WHERE (tax\_id=1 or tax\_id=2)
    and (carcassweight>=300 and carcassweight<=400)
    and (owner='PL' or owner='DE')
UNION
  SELECT breed\_id, mcname, NULL, NULL, NULL FROM breeds
  WHERE (owner='FR') and (tax\_id=3)
UNION
  SELECT breed\_id, mcname, NULL, tax\_id, dailygain FROM breeds
  WHERE (dailygain>=24 and dailygain<=56) and not((tax\_id=1 and so on ...));
```

In result we get a following view:

breed_id	mcname	country_id	tax_id	dailygain
33	Polish Red	50000091	1	NULL
45	Angler	50000009	1	NULL
67	Wollschwein	50000009	2	NULL
56	Pulawska	50000091	2	NULL
23	Duck de la France	NULL	NULL	NULL
78	Lanka	NULL	5	350
24	Florina	NULL	6	315

Table 4.12: View for the breeds table

The symbolic schema of reading data is shown on Figure 4.4.

4.8 Grouping access rights

All access rights in the system are granted to the user by the groups. The group is a bunch of access rights which can be assigned to the one or more user. Each

⁹NOT expression can help prevent views before duplicated records

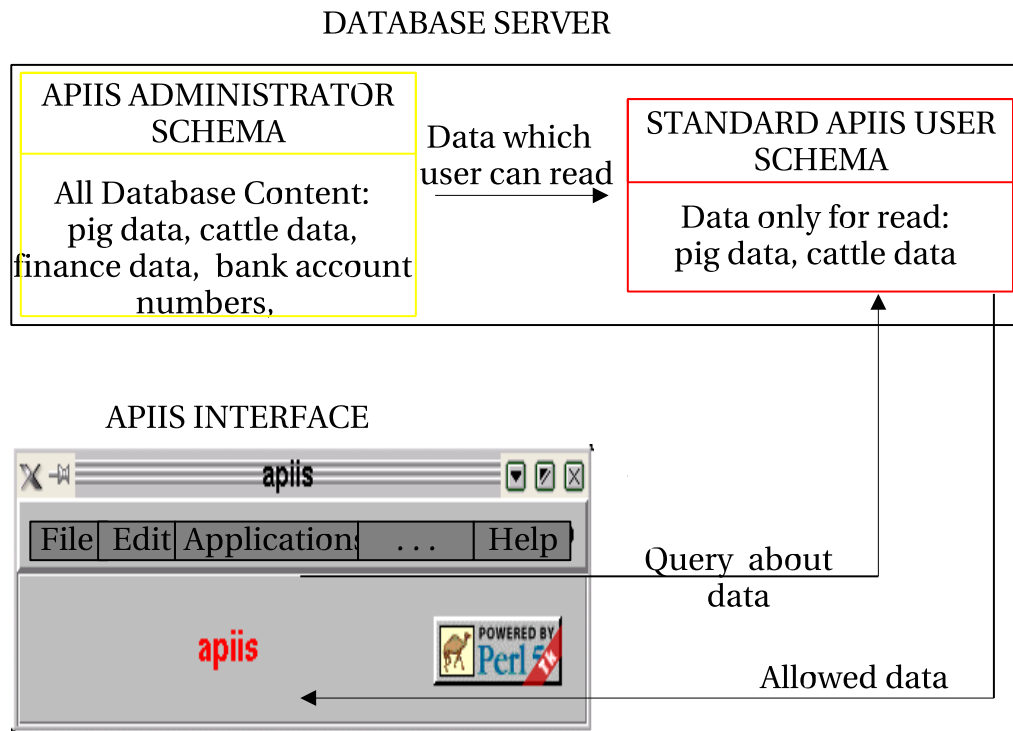


Figure 4.4: Reading data from the database

group can be structured from roles or other groups. The important thing is that these two elements can not be mixed in the one group. The main role group definitions are kept in the database table (AR_Groups)¹⁰.

group_id	group_name	group_type	group_content	group_desc
1	system_task_administrator	st_group	Roles	description
2	database_administrator	dbt_group	Roles	description
3	breeder	dbt_group	Groups	description

Table 4.13: Groups

¹⁰The table was already presented in the section 4.5

4.8.1 Grouping roles

Each role is allocated at list in the one group. The relations between the roles and role groups are kept in the separate table (AR_Role_Groups). The role is allocated for the group by the administrator. If we want to add the role to some group, first we have to check the group type (the type of the role has to be the same type like the type of the group) and the group content (role can be added to the group which contains roles definition - not other groups). If these two requirements are agreed then we have to investigate that the new role can cooperate with the other roles which are currently defined in this group. This process is done automatically on the basis of the role constraints defined for the roles. These constraints qualify which roles can not be used in the same time in the one role group. The constraints for the roles are stored in the different table (see section 4.13, figure 4.8): AR_Role_Constraints) than the constraints for the groups (see section 4.13, figure 4.8): AR_Group_Constraints).

role_cons_id	role1_cons_id	role2_cons_id
1	1	2

Table 4.14: Constraints for the roles

The fields role1_id and role2_id are foreign keys to the roles table (AR_Roles table). The algorithm, which verifies the roles, takes from the group (to which we want to add the role) the current list of its roles. The values from the list are set together one by one with the id of the new role. Each couple of values is used as a condition for the WHERE clause in the following SQL statement:

```
SELECT role_cons_id FROM ar_role_constraints WHERE
(role1_cons_id='existing_role' and role2_cons_id='new_role') or
(role1_cons_id='new_role' and role2_cons_id='existing_role')
```

For each couple of roles one SELECT is executed. When all combination of roles are positively verified (no results for each combination) then the role can be appraised to the group. If there is a result for some union then this means that there are some constraints and role can not be added to the group. The algorithm is not stooped in this point and it just go through the all combinations. All results are collected and then they are showed to the administrator. The administrator has clear picture which roles are in the conflict with the new role.

4.8.2 Grouping groups

The groups can be also assigned to the other groups. This can be done only if the content of the group to which we want to add new group is defined as "Groups" and the types of the groups are the same. The relations between the groups are kept in the separate database table (see section 4.13, figure 4.5): AR_Group_Groups) where we define the group_id from higher level (parent) and group_id from lower level (child). In this table the unique key is defined on both of the columns. The important rules are that the group can not be ascended to itself and also that there is no possibility to create the same combination of groups but with different order of columns (the group ids changed between the

columns).

If we want to add the group to the other group, we have to be in right with the condition presented above. Then we have to check the group constraints (checking that the new child group can cooperate with the other already defined child groups). This constraints are stored in the same table where the constraints for the assigning user to the groups are defined (AR_Group_Constraints - Table 4.4). The difference is only in the relation type, her it is defined as "group-group".

In this case the algorithm takes from the parent group the current list of its child groups. The existing children are set together one by one with the new child. Each couple of values is used as a condition for the WHERE clause in the following SQL statement:

```
SELECT group_cons_id FROM ar_group_constraints WHERE
  ((group1_cons_id='existing_child_group' and
    group2_cons_id='new_child_group') or
    (group1_cons_id='new_child_group' and
    group2_cons_id='existing_child_group')) and
    (group_cons_type='group-group-cons')
```

For each couple of role values one SELECT is executed. When all combination are positively verified (no results for each combination) then new child group can be appraised to the parent group. If there is a result for some union then this means that some constraints are defined and new group can not be added. The algorithm is not stooped in this point and it just go through the all combinations. All results are collected and then they are showed to the administrator. The administrator has clear picture which existing groups are in the conflict with the new child group.

4.9 Specifying constraints for the grouping

In the previous sections we specified the three types of constraints which are use in the grouping:

1. user-groups-constraints - checking if the user can be ascribed to the new group with his current aggregation of groups
2. group-groups-constraints - checking if the group can be defined as a part of other group.
3. role-constraints - checking if the role can be added to the group

The manner of adding new constraints for each of this category is very similar. At the beginning we have to check that the new constraints will be valid for the current definitions allocated for the user and groups (f.e. one of the user is assigned to two groups which we want to exclude). Thus the one of the following statements have to be executed:

```
SELECT user_id FROM ar_user_groups WHERE group_id='first group
  id' or group_id='second group id'
```

This SELECT is executed for the first type of constraints. In the WHERE clause we put these group ids for which the new constraint will be defined. As a result we get a users which are attributed for these groups.

```
SELECT hl_group_id FROM ar_group_groups WHERE ll_group_id='first  
group id' or ll_group_id='second group id
```

The SELECT is executed for second type of constraints. In the WHERE clause we put these group ids for which we want to define new constraint. SELECT returns these parent groups which have such child groups defined.

```
SELECT group_id FROM ar_role_groups WHERE role_id='first role id'  
or role_id='second role id
```

The last SELECT is executed for the third type of constraints. In the WHERE we put the role_id for which we want to define new constraint. Returned results give the information about groups to which these roles are assigned.

After this when the SELECT is executed, the algorithm checks if there are any duplicates in the returned results (two the same user, two the same groups). If the duplicates are presented for one of this SELECT then they are returned as an one conflicts list. In such case the new constraints can not be added because it causes a contradiction in the current definitions. The administrator first has to change these conflict definitions and then this constraints can be introduced.

4.10 Further developing

4.10.1 Checking the login time and the current status of the users

The method of checking the login time for the user is needed to prevent system before unclosed session. There are three columns in the users table which are used by this method: `session_status` column which is fill in during the logging (the flag of this column is set as ACTIVE), `last_login` column which is fill in by the logging timestamp and `last_activ_time` column which is updated during the user session. The last column is updated by the actual timestamp everytime when the user executes some action on the database. In the same time algorithm checks also the activation time for the all others users and compares it with the actual time. If the difference for some user is greater then defined timeout then the user session is closed. The timeout should be defined as a global value in the configuration file.

4.11 Remarks

Implementation of the Security System is made in the Perl Programming Language.

4.12 Bibliography

- [1] <http://www.postgresql.org/docs/7.4/static/sql-createschema.html>
- [2] David F. Ferraiolo, Ravi Sandhu, Serban Gavrila, D. Richard Kuhn, Ramaswamy Chandramouli, Proposed NIST Standard for Role-Based Access Control, (ACM, 2001)
- [3] <http://csrc.nist.gov/rbac/>
- [4] <http://en.wikipedia.org/wiki/SSH>

4.13 ERD diagrams

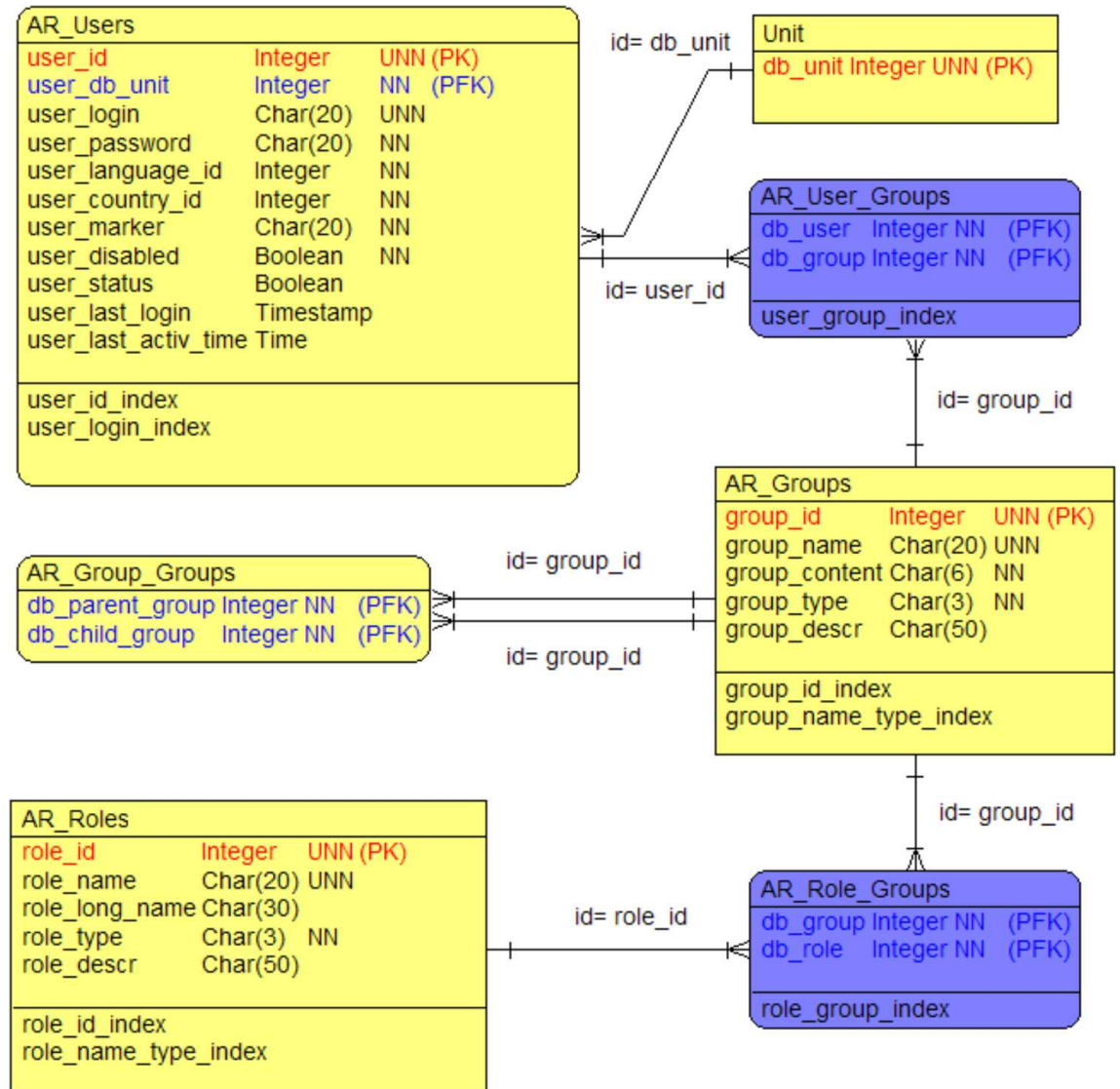


Figure 4.5: Users-Groups-Roles

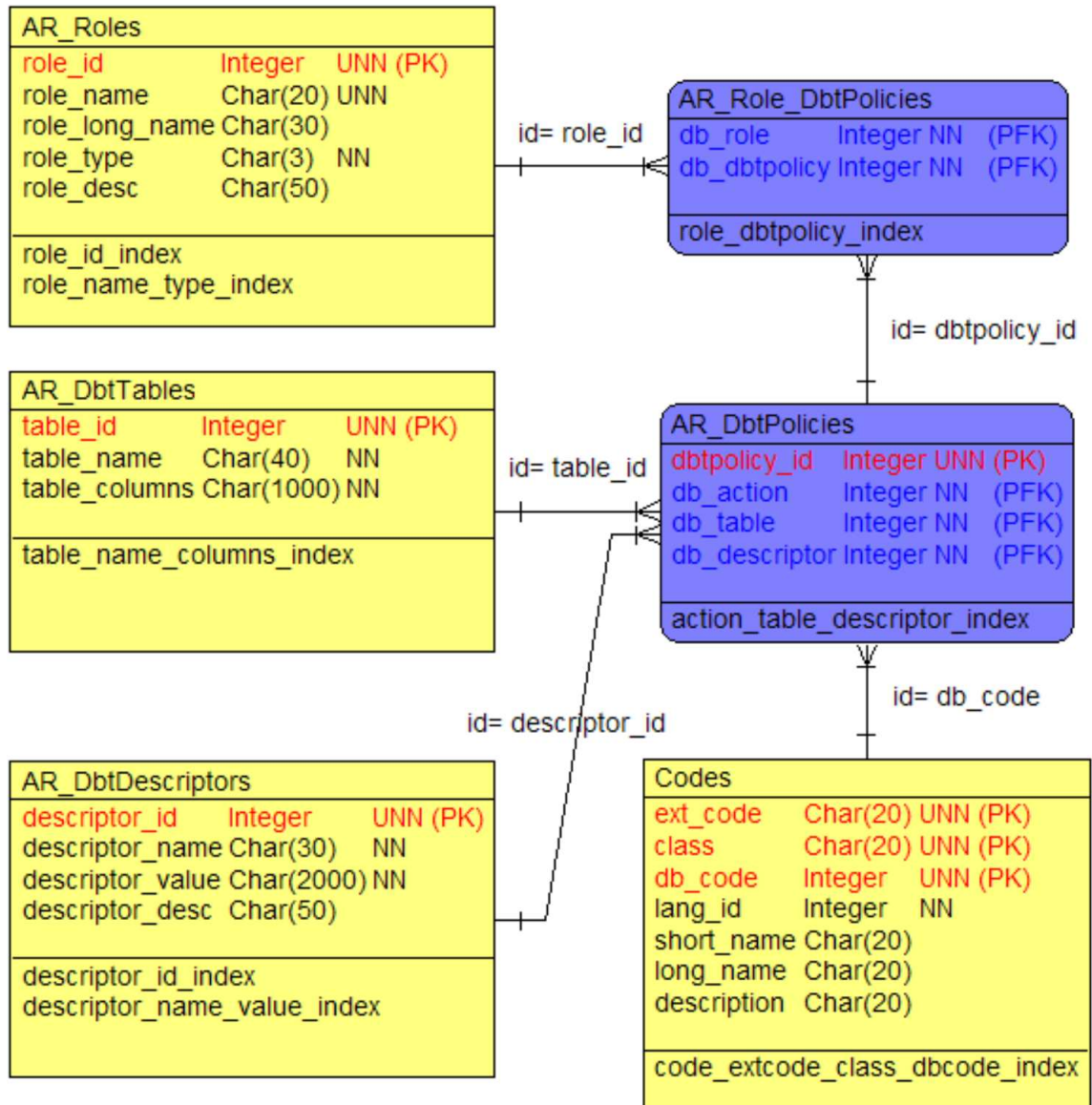


Figure 4.6: Access rights for the database tasks

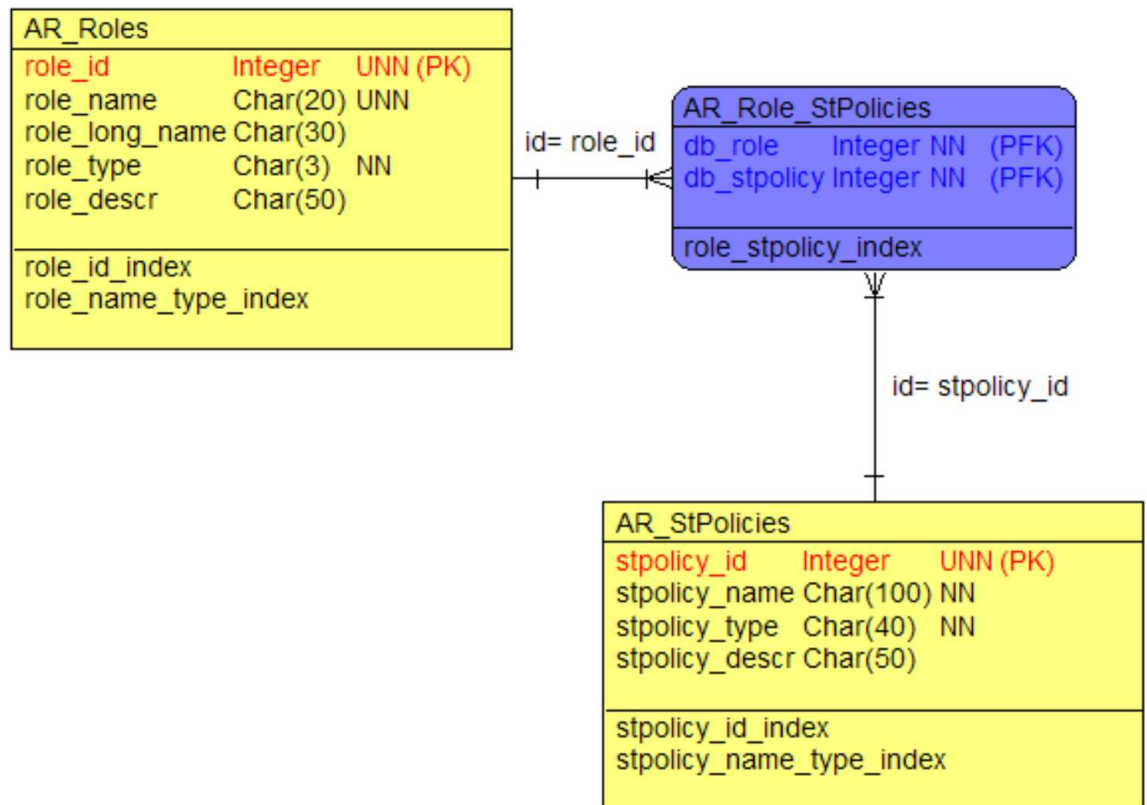


Figure 4.7: Access rights for the system tasks

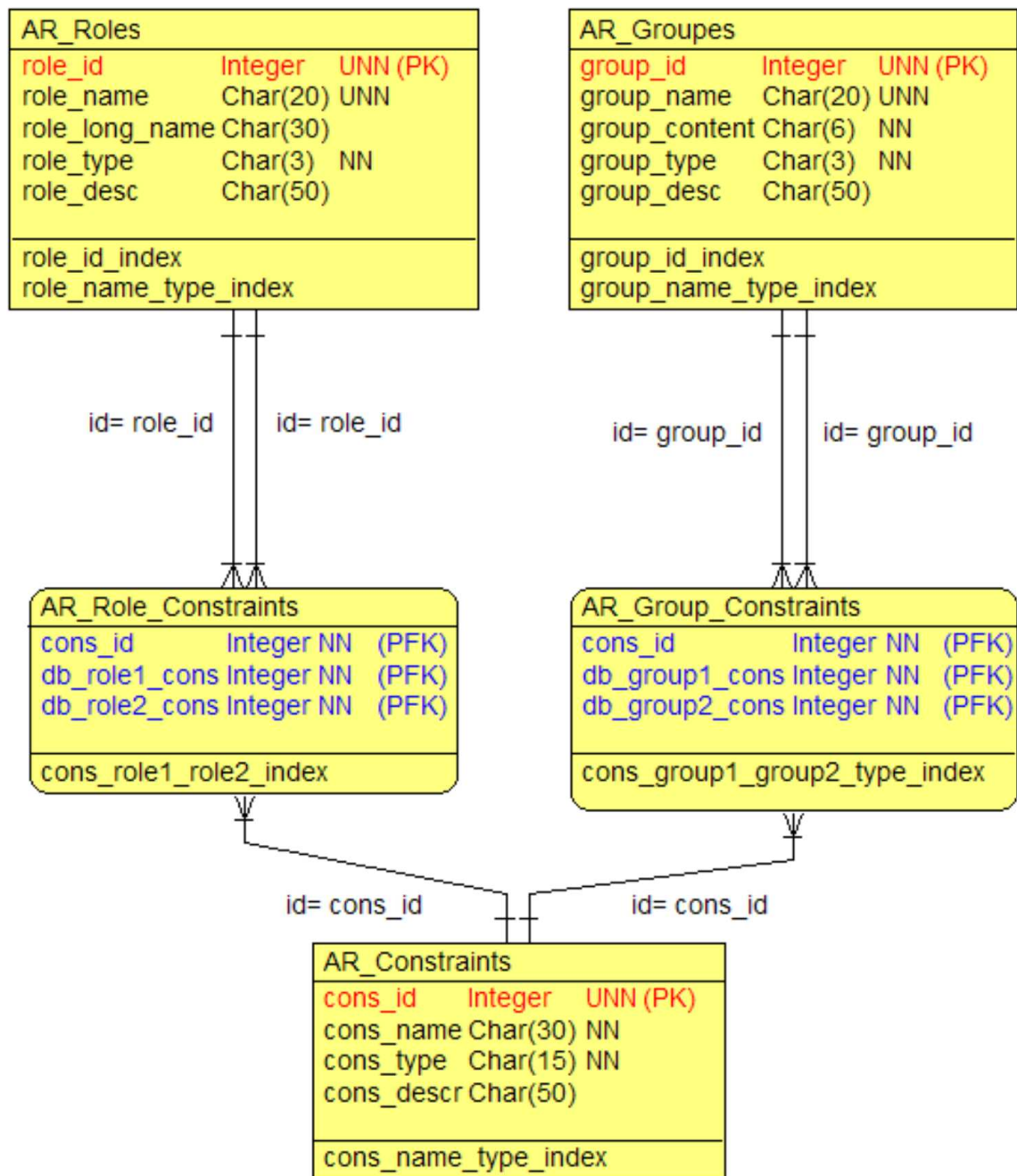


Figure 4.8: Constraints

Chapter 5

Synchronization of Database Content

5.1 Database structure

Here are described all elements of the database structure that are related or dedicated to the synchronization process.

5.1.1 Additional fields

In each table there are three meta-fields that are related to the synchronization: “**guid**”, “**owner**” and “**version**”. These fields are set automatically by the system and contain respectively the unique record number, the record owner (i.e. in EFABIS this is the node name) and the record version number.

There is also an “ordinary” boolean field in each table called “**synch**”, that is set by user and that determines if the record is targeted for synchronization. This field is required, because there is no generic way to distinguish between records from one and the same table, that have to be synchronized and that do not.

5.1.2 Additional tables

There are three “system” tables used explicitly for synchronization purposes: “**nodes**”, “**sources**” and “**targets**”. These tables describe all nodes in the system and the data flow from one node to another.

- nodes - contains all node names and IP addresses

```
CREATE TABLE nodes (  
    guid          int4,          -- global identifier  
    nodename      text,          -- node name  
    address       text,          -- node ip address  
    last_change_dt timestamp,    -- Date of last change, automatic timestamp  
    last_change_user text,      -- User who did the last change  
    dirty         bool,         -- report errors from check_integrity  
    chk_lvl       int2,         -- check level
```

```

    owner          text,          -- record class
    version        int4,          -- version
    synch          bool           -- is record targeted for synchronization
);
CREATE UNIQUE INDEX uidx_nodes_1 ON nodes ( guid );

```

- sources - contains information about the data elements this node expects from the others

```

CREATE TABLE sources (
    guid           int4,          -- global identifier
    source         text,          -- source node
    tablename      text,          -- table name
    class          text,          -- owner node
    columnnames    text,          -- columns
    last_change_dt timestamp,     -- Date of last change, automatic timestamp
    last_change_user text,        -- User who did the last change
    dirty          bool,          -- report errors from check_integrity
    chk_lvl        int2,          -- check level
    owner          text,          -- record class
    version        int4,          -- version
    synch          bool           -- is record targeted for synchronization
);
CREATE UNIQUE INDEX uidx_sources_1 ON sources ( guid );

```

- targets - contains information about the data elements this node distributes to others

```

CREATE TABLE targets (
    guid           int4,          -- global identifier
    target         text,          -- target node
    tablename      text,          -- table name
    class          text,          -- owner node
    columnnames    text,          -- columns
    last_change_dt timestamp,     -- Date of last change, automatic timestamp
    last_change_user text,        -- User who did the last change
    dirty          bool,          -- report errors from check_integrity
    chk_lvl        int2,          -- check level
    owner          text,          -- record class
    version        int4,          -- version
    synch          bool           -- is record targeted for synchronization
);
CREATE UNIQUE INDEX uidx_targets_1 ON targets ( guid );

```


Chapter 6

Internationalization in APIIS

This chapter deals with the localization (internationalization?) aspects and the design in APIIS. Here, we distinguish two parts:

- localization of interfaces
- localization of database content

6.1 Localization of Interfaces

Helmut: hier kennst Du Dich aus.

6.2 Multilanguage Handling of Database Content

In the EFABIS database we have much database content that needs to be available in a number of languages. These are for instance breed descriptions, codes like male and female. With national database feeding their data into the regional database at the EAAP breed descriptions will have to be in the national language, while for the upper levels (EAAP and FAO) these will have to be in one of the official languages. As a result, if data are to be entered for a public record at the national level a national and an international version of the same field will have to be supplied. Operationally, only the international version will go the regional level, while the national information will stay where it is. Being public data, the new record will go up the FAO. From there it will go down to a translator, all via the standard synchronization protocol. The translator will create a new record in a new official language, this will then via the same protocol go down to the regional and finally national level. Thus, there will be private (local) and public (international) data for one entity.

6.2.1 Implementation

6.2.1.1 Database Structure

Starting point is a table that initially contains multilingual content, i.e. text columns that will appear in more than one language. The procedure followed

Table 6.1: master part

breed_id (PK)	male	female
25	2	0
26	0	2

Table 6.2: translation part

breed_id	language_id	shape
25	2	Dlugie krecone rogi
25	1	Long curved horn

is:

1. split the table into a part that contains only columns which do not need translation, i.e. numerical values. This part will keep the original primary key. For the breeds table in EFABIS this will be the breed_id.
2. have another part that contains all those columns that need translation, one row per language. Accordingly, the primary key will be composed from the original primary key (the breed_id in the EFABIS example) and the language. This will be a pointer to languages table where we have ISO language code eg. PL or EN and other language description.

An example is given in table 6.1. This is the "master" with PK breed_id while the second table 6.2 has a unique key consisting of breed_id and language.

Some tables will not be splitted into two parts eg. CODES. Primary key of is defined as concatenated from ext_code, class and lang_id where closing date is NULL. Additional table for language information was added (see table 6.3).

Generally tables which have translatable primary key or part of primary key will not be splitted into two tables. Columns ext_name in naming table and ext_unit in unit table are not translatable.

6.2.1.2 Encoding

Currently, encoding relies on one language only. Extension to a multi language setting will be made through views. Views will mask tables used for current coding/encoding only for one language. Thus views will be created dynamically in user schema. Name of view will be same as table name. Language views will be created after access right views. Thus, in a load object we want to insert the

Table 6.3: language table

lang_id	iso_lang	lang
1	EN	English
2	PL	Polski or Polish?

translation of a German and English text. Assuming that the column 'shape' is encoded in the model file, then the pseudo SQL in a load object will look like:

```

set_lang=>'DE';$lang='DE';
$pseudo_sql[1] =
'INSERT INTO breeds_lang_horns (
    lang,
    breed_id,
    shape
) VALUES (
$lang,
    $breed_id,
    'gerade'
)';
set_lang=>'EN'; $lang='EN';
$pseudo_sql[1] =
'INSERT INTO breeds_lang_horns (
    lang,
    breed_id,
    shape
) VALUES (
$lang,
    $breed_id,
    'straight'
)';

```

The call to the object `set_lang` (Apiis->language?) will set the language for the encoding to either 'DE' or 'EN' and create or recreate appropriate views for coding/encoding.

6.2.2 Meta level

The meta layer will be treated as before. (Zhivko???: The meta level should be based on the view system build for one language.) Probably it will be the only change there.

- Inserting a new record
SELECT \$breed_id FROM breeds WHERE mcname=\$mcname AND country_id=\$country_id
AND tax_id=\$tax_id;
INSERT INTO breeds_horns_master (breed_id,male,female) VALUES (\$breed_id,\$male,\$female);
INSERT INTO breeds_horns_transl (breed_id,lang_id,shape) VALUES (\$breed_id,\$lang,\$shape_pl);
INSERT INTO breeds_horns_transl (breed_id,lang_id,shape) VALUES (\$breed_id,\$lang1,\$shape_en);
- Updating existing record
SELECT \$breed_id FROM breeds WHERE mcname=\$mcname AND country_id=\$country_id
AND tax_id=\$tax_id;
UPDATE breeds_horns_master SET male=\$male WHERE breed_id=\$breed_id;
UPDATE breeds_horns_transl SET shape=\$shape_en WHERE breed_id=\$breed_id AND
lang_id=\$lang1;

All statements have be wrapped in one transaction block.

6.2.3 Access rights

Again the Access rights system should be revisited, but probably will stay the same. There should be only mechanism for consistency of the user rights, i.e. if he can enter the most common name (which is in the master table because is not translatable), he has to be able to enter the language of the most common name (which is in the translation table).

6.2.4 Synchronization

Since the structure will be the same on all levels and we have clear mechanism to mark the records to be synchronized the synchronization will remain the same. All records to be synchronized will be marked manually by the user.

6.2.5 Outputs

Outputs could be routed via views which can be created (automatically?) for each language. It would simply present the master and translation table as one for a given language via a join on the primary key of the master with the primary key and language (which should be a unique composite index) for the translation table.

6.3 International Character Sets

Localized languages require for their presentation a corresponding character set. Therefore, we do not only need to take note of the correct character set that goes with a language but also need to be able to use them appropriately. It is the problem to work with many different character encodings. Two character encodings can use the same number representation for two different characters, or use different numbers for the same character. Any given computer (especially servers) needs to support many different encodings. Now whenever data are sent between different encodings or platforms there is a risk of data corruption when we don't use appropriate character encoding. The best way is to use one character encoding which can cover all languages used in a system. The best way here is to use UNICODE standards.

6.3.1 UNICODE

Unicode and ISO/IEC 10646 are coordinated standards that provide code points for characters in almost all modern character set standards, covering more than 30 writing systems and hundreds of languages, including important modern languages. All characters in the largest Chinese, Japanese, and Korean dictionaries are also encoded. Unicode 1.0 was released in October 1991, and 4.0 in April 2003. A Unicode character is an abstract entity. Unicode provides a unique number for every character, no matter what the platform, no matter what the program, no matter what the language.

6.3.2 Support for UNICODE

UNICODE is supported in many operating systems, all modern browsers, database and programming environments. PostgreSQL database has support of UNICODE (from version 7.1). After using UNICODE as a database character encoding we can store any language in database text field, if we want we can generate outputs in other encodings eg. LATIN2 because Postgres has support for automatic characters encoding but it is better to use one character encoding everywhere. Perl is also supporting Unicode characters (from version 5.6, but more complete support is in Perl from version 5.8). Perl should automatically work with strings encoded in Unicode correctly. If we want also use variables

name encoded in Unicode we need to specify this with Perl pragma:

```
use encoding 'utf8';
```

after that we can use variables name with national characters. Perl characters representations is hidden from user. Perl scripts should work with unicode data same as with data encoded for eg. latin but it depends on operating system environment. Probably we need examine regular expressions for unicode compatibility because we now will have much wider characters set then in eg. LATIN encoding. In some places we need accept not only [A-Za-z] characters but also some national special characters as German ö, ä, ü etc. with [:alpha:] (see perlre documentation for more). For user interface it should be no problem to use unicode for web interface, also a Tk interface is working with unicode (from version 8.00) only need correct font to show all characters.

6.3.3 Locales and fonts

For proper working of Unicode settings in database or in Perl script we also need to use LOCALE settings. Locale are important when we are sorting and comparing strings with national characters. By default Perl is ignoring locale settings if we want to use it we need to specify it:

```
use locale;
```

This is needed if we for eg. want to change character case with lc or uc function in Perl. If we don't specify this that Perl will not change case correctly for all characters in our string. Most of Perl operators don't need to know about characters in string. If there are utf8 characters in is working with utf8 characters. Generally is hiding internal representation of characters from outside world.

Unicode is platform independent but if we want to use national characters on some operating system we need to have appropriate fonts which can show all characters described in Unicode. It is mostly important for languages like Chinese, Japanese, Korean, Arabic etc.

- how do we handle latin based character sets?
- what about others like cyrillic?
- even more complicated: what about Arabic, Chinese, Japanese etc?
- do we have to go UNICODE?
- what can be put in a field of type text in the database? Using a local keyboard and the appropriate character set, can we store Bulgarian in one row and French or Chinese in others?

6.4 The Core

6.4.1 Apiis.pm

SYNOPSIS

```
use Apiis;
Apiis->initialize( VERSION => '$Revision: 1.4 $' );
```

DESCRIPTION

initialize is the primary method for executables to load the Apiis system. It does basic checking, creates, and exports the \$apiis object into the main namespace.

To avoid numerous nasty error messages you are strongly advised to start your program with this BEGIN block:

```
BEGIN {
  use Env qw( APIIS_HOME );
  die "\n\tAPIIS_HOME is not set!\n\n" unless $APIIS_HOME;
  push @INC, "$APIIS_HOME/lib";
}
```

This catches errors due to an unset APIIS_HOME environment variable and adds \$APIIS_HOME/lib to your library path to find the Apiis modules.

SUBROUTINES

initialize **initialize** loads Apiis::Init, creates a new Apiis::Init object and assigns it to the global variable \$apiis, which is exported by default. Also exported is the global subroutine __() for nationalisation of the code.

initialize currently takes one (hash) argument:

```
Apiis->initialize( VERSION => '$Revision: 1.4 $' );
```

It propagates the cvs version as the program version and can be retrieved with \$apiis->version.

Author

Helmut Lichtenberg <heli@tzv.fal.de>

6.4.2 Apiis::Init – Basic initialisation object for the complete APIIS structure**SYNOPSIS**

```
our $apiis = Apiis::Init->new(
  version      => $version,
  programname => $programname
);
```

This is the basic module for spreading the APIIS configuration during runtime. It is invoked automatically if you run the common initialisation block which includes \$APIIS_HOME/lib/apiis_init.pm. You can access this basic object via the global variable \$apiis.

DESCRIPTION

Apiis::Init creates an internal structure and public methods to access this structure.

Public and internal methods are:

INTERNAL METHODS

new (mostly internal) new creates the object where we usually refer to as \$apiis.

_init (internal) _init does the main initialization and creates the internal structure for:

APIIS_HOME os _user version programname date _format entry _views reserved _strings language codes _table browser fileselector

This is done by querying parameter from the operating system (username) and the user environment (APIIS_HOME). APIIS_LOCAL is set after a certain project is selected and the model file is joined into \$apiis.

The main resources for this basic structure are the configuration files \$APIIS_HOME/etc/apiisrc and later \$APIIS_LOCAL/etc/apiisrc.

_get_user_from_os (internal) The username \$apiis->os_user is determined by the operating system. This is mainly needed for initial log messages, who started the program.

projects (public) Returns the names of the projects defined in \$APIIS_HOME/etc/apiisrc.

project (public) Returns the \$APIIS_LOCAL path for a specific project and is therefore mostly redundant with \$apiis->APIIS_LOCAL().

Example:

```
$local_path = $apiis->project('ref_breedprg');
```

formpath (public) Returns the default path for a specific project where the form definitions are stored, usually at \$APIIS_LOCAL/etc/forms. This default location is set during join_model.

It can be set to a different value with:

Example:

```
$apiis->formpath( './forms' );
```

l10n_init (public) l10n_init does the localisation from Apiis::I18N::L10N. The language is passed as input parameter.

The failure handler for Locale::Maketext is set to return the untranslated english string (default language).

Also the defined projects translations table is imported into the l10n schema.

Input: language

Output: none

l10n_import (public) l10n_import imports an additional lexicon. This is usually done by l10n_init. In case you want to load another lexicon, use l10n_import.

Input:

1. language
2. file

Output: none

Example:

```
$self->l10n_import( $lang, $lexicon ) if -f $lexicon;
```

`__()` After initialization of the language handle `$lh`:

```
$lh = Apiis::L10N->get_handle( $apiis->language );
```

you could write for localising text:

```
print $lh->maketext('Just another Perl hacker'), "\n";
```

To make it more convenient I created a shortcut wrapper around this, the subroutine `__()`. So you can write:

```
print __('Just another Perl hacker'), "\n";
```

Note: I must use an anonymous subroutine to have access to `$lh`. See 'Programming Perl', p. 976 for error message: 'Variable "\$lh" will not stay shared'.

Note2: The bare underscore `_` is treated specially, as it is always forced into the package `main` (like `$_`, `@_`). See "Programming Perl", p. 591. So we don't have to export it.

Note3: The single underscore `_` produced errors several times when it clashed with the "Perl special filehandle used to cache the information from the last successful `stat`, `lstat`, or file test operator". ('Programming Perl', p. 657). This global underline subroutine is used e.g. in the CPAN or CGI modules. So it's better to *not* use `__()` for localisation. Preferred shortcut now is `__()`. (5. Aug. 2004 - heli)

`_add_obj (internal)` `$self->_add_obj` is used to mount an additional object structure into the `apiis` core structure. An example is the addition of the model file information under `$apiis->Model`.

usage:

```
$self->_add_obj(
    Model => [ $mod_obj ],
    caller => [ $package, $file, $line ]
);
```

PUBLIC METHODS

`$apiis->[os_user | APIIS_HOME | version | programname | date_format | entry_views | reserved_strings | codes_table | browser | fileselector | use_filelog | filelog_filename | use_syslog | syslog_facility | use_sql_logging | sql_logfile | sql_log_dml_only | node_name | node_ip | sequence_interval | multilanguage]` (all public) These public methods provide an interface for the user to access the internal structure.

They are readonly and usually return a scalar value except `entry_views` and `reserved_strings`.

`$apiis->entry_views` returns a hash reference with the table names as keys and the according entry views (which only contain active records of this table) as values:


```

codes => entry_codes
unit => entry_unit
transfer => entry_transfer

```

`$apiis->reserved_strings` returns a hash reference to the names and values of the reserved strings for data entry:

```
v_concat => ' >=< '
```

(One problem here could be the intended blanks as part of the delimiter. Maybe they got lost by reading the config file with `Config::IniFiles`.)

\$apiis->[language] (public) language is a public read/write method. Initially it's populated by the `apiisrc` configuration files, but it can be changed during program execution. When you set a new language, the old one is returned:

```
my $oldlang = $apiis->language( <newlang> );
```

\$apiis->[date_order | time_order | extdate2iso | iso2extdate | ext-time2iso | iso2exttime | date_parts | time_parts | isodate | isotime | date_sep | time_sep | date_conf_err | time_conf_err] (public)
The `Apiis` default format for date and time accords to the widely accepted ISO 8601 standard. Have a look at

<http://www.cl.cam.ac.uk/~mgk25/iso-time.html>

for a good summary or other resources for detailed descriptions. You are strongly encouraged, to also use ISO 8601 date formats in your software.

date_order returns the initially in `apiisrc` defined order of the date as a scalar string. You can set the date format during program execution (e.g. when you batch process several data streams) in the following syntax:

```

my $oldformat = $apiis->date_order(
    order => 'DD.MM.YYYY',
    sep   => '.',
);

```

The two required parameters are the order of the parts and the separator. The string to define the order has the following limitations:

- only the separator and the capital letters Y, M, and D are allowed.
- the year has to be specified in the 4 digit form YYYY to avoid ambiguity.
- the day (DD) and month (MM) formats have 2 digits each.
- a valid order string with separators therefore must have the length of 10 characters.
- a valid order string without separator must have the length of 8 characters.
- for year, month, and day values only digits are allowed.

Example without separator:

```
my $oldformat = $apiis->date_order(
    order => 'YYYYMMDD',
    sep   => '',
);
```

If you want to set **date_order** to new values, it returns a reference to the hash of the previously configured parameters order and sep. You thus can reset the old date format with:

```
$apiis->date_order( %$oldformat );
```

If the chosen date order accords to ISO 8601 (YYYY-MM-DD) the status flag `$apiis->isodate()` is set to 1, otherwise its 0. The same applies to the time order (hh:mm:ss) and `isotime()`.

Another flag **date_conf_err()** is internally used to mark a bad date format configuration and as a result of it skip all date tests.

If you really have to parse dates on your own you can get the separators (besides the format string with **date_order**) by invoking:

```
my $d_sep = $apiis->date_sep();
my $t_sep = $apiis->time_sep();
```

date_parts() is a readonly public methods that returns an array (or an arrayref, depending on the invoking context) of the configured parts of the date format in the correct order (e.g. ["YYYY", "MM", "DD"]).

This method is mainly usefull in internal date calculations.

extdate2iso converts your external date format into the internal ISO 8601 format. It additionally checks, if the passed date is valid.

In scalar context, a formatted date string is returned. In list context, you get the date parts in the shown order:

```
Example:
$apiis->date_order( order => 'DD.MM.YYYY', sep => '.' );

# scalar context;
my $ext_date = '11.2.2005 13:37:00';
print $apiis->extdate2iso($ext_date), "\n";
# prints: 2005-02-11 13:37:00

# list context:
my ( $year, $month, $day, $hour, $minute, $second )
    = $apiis->extdate2iso($ext_date);
```

The same return schema for scalar and list context applies to **exttime2iso**, **iso2extdate**, and **iso2exttime**.

Note, that also the **iso2extdate** and **iso2exttime** methods keep this order in list context. It does not make sense to make them return in the configured external order as the list context is useful for programming purposes and a changing order would force you to parse the configuration. And this is not, what you want.

substitute_env (internal) Does some postprocessing for special cases (substitution of APIIS_HOME and APIIS_LOCAL with their values).

The value to check for substituting is passed as a reference so that substituting is done in place:

```
$self->substitute_env( \ $val_to_substitute );
```

It doesn't matter if there is a dollar sign \$ in front of APIIS_HOME and APIIS_LOCAL or not.

_join_user (internal) \$apiis->_join_user takes a hashref with a User object (required) and verifies this user against the database. If it's a valid user, his data gets mounted into the \$apiis structure as the User object.

Example: \$apiis->_join_user({ userobj => \$user_obj });

exists_user (public) \$apiis->exists_user returns 1 if the User object is already mounted into the \$apiis structure, 0 otherwise.

use_filelog/use_syslog/use_sql_logging (public) These methods mainly reflect the settings in apiisrc. They are read/write to enable changing these values in rare cases, e.g. when running check_integrity, where logging make only little sense.

syslog_priority/filelog_priority (public) syslog_priority is read/write although it mostly won't be overwritten. But in some cases you may want to switch the logging level for a certain part of the code to e.g. 'debug', while other parts stay at e.g. 'warn'. The allowed priorities are debug, info, notice, warn, warning, error, err, crit, alert, emerg, panic in this order (err = error, warn = warning, emerg = panic).

If syslog_priority is set with

```
my $oldvalue = $apiis->syslog_priority('debug');
```

it returns the old value of syslog_priority. You then can reset it with

```
$apiis->syslog_priority( $oldvalue );
```

Otherwise it returns the current value of syslog_priority.

The same applies to **filelog_priority**.

log_priority (public) **log_priority** is write-only and sets the values of syslog_priority and filelog_priority to the same value which is passed as the argument. This is mainly a development help as you don't know if the configuration is just set to syslog or filelog.

debug (public) **debug** returns 1 if the debug level is set, 0 otherwise. Any true input value sets \$self->debug to 1, any false value to 0.

debug can be used to query or set a debug flag, which can be used to prevent the expensive invocation of \$apiis->log on debug level. This flag depends on the settings of filelog_priority and syslog_priority. If either of them is set to 'debug', \$self->debug always returns 1, even if you pass 0 to it. If you set \$self->debug(1), filelog_priority and syslog_priority keep their values;

log (public)

```
$apiis->log('warn', "Cannot open file: $!");
```

or

```
$apiis->log('warn', 'Cannot open file: %s', $!);
```

`log()` is the interface to the `syslog` utility. It takes as first input parameter the `syslog` priority, at which it shall be printed into the system log files (debug info notice warn warning error err crit alert emerg panic). All levels below `$apiis->syslog_priority` are suppressed, all of `$apiis->syslog_priority` and above are sent to `syslog`.

As an addition it can also log the sql statements into a file for basic database recovery. If the passed priority is of type 'sql' like in

```
$apiis->log('sql', $sqltext);
```

and `use_sql_logging` is set to a true value in `apiisrc`, the `sqltext` will get logged into the configured `sql_logfile` together with a timestamp, `dabasename`, and `username` (in a separate line with a sql comment). After a defined backup state you simply have to run this file through your favorite frontend to the database to recover the current state. If `sql_log_dml_only` is true in `apiisrc`, select statements are not logged. Messages of priority 'sql' are not passed to `syslog`.

status (public) `$apiis->status` returns a general status which is accessible everywhere and at any time during execution. A status of 0 means success, all true values indicate an error.

If you pass a parameter this will set the status to this value.

running_check_integrity (public) `$apiis->running_check_integrity` is a simple switch that has to be set in the program `check_integrity`. Some checks on record level have different behaviour (less checks) if they are invoked by `check_integrity`.

check_status (public) Checks `$apiis->status` and prints errors (if any). Optionally dies above a certain severity level and ignores errors below a certain security level.

Input parameter can be a hash with the keys:

- **die** – you can pass a level of severity to let the program die at this point and all levels above (in severity).
- **ignore** – below this level of severity the error messages are ignored

check_status returns the boolean value of the status stored in `$obj->status()`.

Example:

```
$apiis->check_status(
    die => 'CRIT',
    ignore => 'INFO',
);
```

errors (public) \$apiis->errors returns the stored errors as an array of objects or an array reference, just as requested by the caller. If new errors are stored, errors() returns the error id(s). If you store one error object, the error id of this error is returned as a scalar. If you store an array of error objects, an array or arrayref of the error ids of these error objects is returned in the order of the error objects.

Examples: my \$err_id = \$apiis->errors(\$error_object); my @err_ids = \$apiis->errors(@error_objects); my \$err_ids_ref = \$apiis->errors(@error_objects);

error (public) \$apiis->error takes as parameter an error id and returns the error object for this id. This enables you to write code like this:

```
$apiis->error(3)->print;
$apiis->error(4)->severity('CRIT');
```

If you pass an invalid error id, an error object is created and passed back to the caller.

del_errors (public) \$apiis->del_errors deletes all error objects.

del_error (public) \$apiis->del_error takes as parameter an error id and deletes this error object from the \$apiis->errors array. Example:

```
$apiis->del_error(3);
```

If you pass an invalid error id, an error object is created, added to \$apiis->errors and additionally passed back to the caller.

localtime (public) \$apiis->localtime provides you with an unformatted timestamp. Usually this is not used. The preferred methods are \$apiis->today and \$apiis->now as they convert the date/time to the localized format.

\$apiis->localtime returns a list of parameters. Example:

```
my ($year, $mon, $mday, $hour, $min, $sec)
    = $apiis->localtime;
```

today (public) \$apiis->today returns a formatted string of the current day.

now (public) \$apiis->now returns a formatted string of the current day and time. For internal use it accepts an input parameter

```
$apiis->now( format => 'today' );
```

to return only the day without time. This is the whole magic behind \$apiis->today. :^)

join_model (public) \$apiis-join_model("modelfile")> mounts all informations of the model file into the core apiis structure and provides methods to access them.

As required input you have to provide the key 'userobj'. The value must be a valid User-object.

Example:

```
$apiis->join_model('breedprg',
  userobj => $user_obj,
);
```

join_model creates an Apiis::Model object and passes it to `_add_obj`. With the key 'Model', the model object is passed as the first and only element of an anon array reference.

Besides the model file name there is another (hash) parameter 'database' to **join_model**.

With 'database => 0', the model file will be joined into \$apiis without connection to the database. For later joining the database into \$apiis, use the public method \$apiis-join_database>.

Using **join_model** without connecting to the database will be used in quite rare cases. One usefull operation will be when you want to drop the complete database during basic initialisation. In this case you have to provide some dummy User object like:

```
require Apiis::DataBase::User;
my $dummy = Apiis::DataBase::User->new(
  id      => ($apiis->os_user || 'nobody'),
  password => 'nopassword',
);

$apiis->join_model('breedprg',
  userobj => $dummy,
  database => 0,
);
```

exists_model (public) \$apiis->exists_model returns 1 if the model file is already mounted into the \$apiis structure, 0 otherwise.

_join_database (internal) \$apiis->_join_database initializes the database access.

It adds the newly created Apiis::DataBase::Init object into the existing \$apiis-tree with the key 'DataBase':

join_database (public) \$apiis->join_database is simply a public wrapper for `_join_database`.

The public method `join_database` is usually not needed as `join_model()` automatically joins the database into \$apiis. For some rare cases (e.g. initial creation of database), you can `join_model()` without connection to the database by passing the parameter 'database => 0'.

So \$apiis->join_model('breedprg', database => 0); \$apiis->join_database; is equivalent to \$apiis->join_model('breedprg');

exists_database (public) `$apiis->exists_database` returns 1 if the database initialisation is already done, 0 otherwise. The existence of the database object does not necessarily include the database connection. If you invoke `join_model` with the parameter `'database => 0'` the database object is created without connecting to the database. This is needed for special cases like `mksql`, where you need the configuration data like the db-specific datatype for the metatypes like `TIMESTAMP` to create the database.

exists_auth (public) `exists_auth` is a boolean switch to show, if the Auth object for authentication/authorisation is joined into the global `$apiis` structure. It is 0/undef, if no Auth object/method exists, 1 otherwise.

get_db_conf (mainly internal) Read the config file for the passed Database from `$APIIS_HOME/etc/apiis/<Database>.conf` and return a hash reference of this structure.

AUTOLOAD (internal) `AUTOLOAD()` catches all invocations of methods, that don't exist. On this level it makes mainly sense for the structural elements `Cache`, `Model`, `DataBase`, `User`, etc. It's difficult to catch them otherwise in expressions like `$apiis->Model->tables`, when `join_model` has failed before and therefore no method `Model()` exists. This case usually produces Error objects, but every developer is free to ignore them.

Currently some more or less useful error messages are generated, printed to `STDOUT` and the process dies. This is not optimal for processes that run in a graphical environment (`Tk`, `Html`) and don't have access to a terminal. But does it make sense to create an Error object if the developer tends to ignore them?

Additionally, the produced error message is stored in the logfile/syslog, if configured.

6.4.3 Apiis::Init::Config mainly ready apiisrc config files

DESCRIPTION

`Apiis::Init::Config` contains internal methods to read the different `apiisrc` files.

METHODS

`_import_apiisrc (internal)` Imports the default `apiis` config file `$APIIS_HOME/etc/apiisrc`.

`_import_apiisrc_local (internal)` Overwrites the defaults from `apiis` `apiisrc` config file with the project specific one.

`_import_user_apiisrc (internal)` Overwrites the project definition from global `apiisrc` config file. This can be used for developers on a multiuser server to point to their private copy of the project tree.

It reads only the `[PROJECTS]` section of `apiisrc`.

__xml2model (internal) **__xml2model** parses the passed xmlfile and returns a reference to a datastructure, representing the model file.

usage:

```
eval { $href = $self->Apiis::Init::Config::__xml2model(
    xmlfile => $filename
);
};
```

This results in a structure like this:

```
$href->{
  general => {...},
  table => {
    <tablename> => {
      struct_type => '...',
      trigger => {...},
      sequence => [...],
      index => [...],
      pk => {...},
      column => {
        <columnname> => {...},
        <columnname> => {...},
      },
      _column_order => [...],
    },
  },
  _table_order => [...],
};
```

Only for internal use.

Note: This method will disappear in the near future when the model file structure and parsing is rewritten.

__get_db_conf (internal) Read the config file for the passed Database from \$APIIS_HOME/etc/apiis/<Database>.conf and return a hash reference of this structure.

\$apiis->__check_date_conf (internal)

Internal routine to run some checks on the configured date format. Sets the `__date_parts` array in case of success.

\$apiis->__check_time_conf (internal)

Internal routine to run some checks on the configured time format. Sets the `__time_parts` array in case of success.

6.4.4 SYNOPSIS

```
$xml_obj = Apiis::Init::XML->new(%args);
$xml_obj = Apiis::Init::XML->new(
    dtd=>$dtd_file,
    xml=>$xml_file,
    gui=>$what_type_of_gui
);
```

6.4.5 DESCRIPTION

XML.pm init a file of configuration written in xml. XML.pm merge definitions from the configuration file and the default values from the dtd-scheme.

Suppositions:

Each xml-element need a unique name over all configuration and subconfiguration files, which will be defined in "Name". Access to each attribute take place with a method in combination with the name of the element:

```
xml:
<PageHeader Name="PageHeader_10">
  <Lines Name="Line_1" Column="1-4" Row="1" LineType="solid"/>
</PageHeader>
-----

code:
$c=$xml_obj->Line_1->LineType
$c is "solid"

$c=$xml_obj->Line_1->Name
$c is "Line_1"
```

Independent of the xml-definition a complete set of methods will be initiate depend on the definition in the dtd-scheme. The default settings come from the dtd-scheme and will overwritten if a the same attribute is defined in the xml-scheme. E.g.

```
dtd:
<!ATTLIST Text
    Name          ID          #REQUIRED
    Content       CDATA       #REQUIRED
    Position      (static|absolute|relative) "relative"
>

xml:
<PageHeader Name="PageHeader_10">
  <Text Position="relative" Name="Text_1" Content="test"/>
</PageHeader>

code:
$c=$xml_obj->Text_1->Position
$c is "relative"
```

Each xml-file has a hierachical structure. XML makes a flat structure.

6.4.6 METHODS

\$apiis->GUI->[fullname | basename | ext | path | gui_file] (all public, readonly)

fullname, basename, ext, path provide the fullname (basename.extension), base-name (without extension), extension, and path of the gui file.

6.4.7 Apiis::Model – methods to access the model file data via the \$apiis structure

SYNOPSIS

```
$apiis->join_model('breedprg');
```

The configuration data of the model file is mounted into the \$apiis structure simply by running the join_model method with the model file name as the only parameter.

DESCRIPTION

This Model.pm module provides an object and the appropriate access methods. With join_model they are passed to Apiis::Init.pm and there with _add_obj added to the global structure

METHODS

new (mostly internal) Apiis::Model->new is mainly invoked by Apiis::Init. The user interface is join_model.

_init (internal) _init does the main initialization and creates the internal structure to keep the model file values.

\$apiis->Model->[fullname | basename | ext | path | db_driver | db_name | db_host | db_port | db_user | db_password | max_check_level] (all public, readonly) fullname, basename, ext, path provide the fullname (basename.extension), basename (without extension), extension, and path of the model file.

The db_... methods reflect the database configurations at the top of the model file.

max_check_level gives you the maximal configured checklevel of this model file, if anybody really needs it.

tables (public, readonly) \$apiis-Model->tables returns the names of the defined tables. If you want an array, it gives you an array of these tables. If you want a scalar, you also get what you want, a reference to the same array.

table (public, readonly)

```
$apiis->Model->table( $tablename );
```

returns an object of Apiis::Model::TableObj for this tablename.

check_level (public, read/write)

```
my $current_level = $apiis->Model->check_level;
my $old_level = $apiis->Model->check_level(2);
... do some work
$apiis->Model->check_level( $old_level );
```

Without an parameter `check_level` returns the current check level. You can change the current check level by passing the new level to `check_level`, which then returns the old check level.

`check_level` also tests, if a passed new level is numeric and does not exceed the maximum defined level in the model file.

Apiis::Model::TableObj – internal package to provide a table object with methods to access a single table and its columns

SYNOPSIS Programming interface:

```
$table_obj = Apiis::Model::TableObj->new( $tablename, $struct_ref);
```

Usage:

```
$table_obj = $apiis->Model->table('animal');
```

METHODS

new (mostly internal) To create the table object, `new()` needs as input the table name and a reference to the datastructure of this table from the model file:

```
$table_obj = Apiis::Model::TableObj->new( $tablename, $struct_ref);
```

The order of the columns in the model file is preserved.

column (public, readonly) `$table_obj->column($col_name)` returns the column object for this column

name (public, readonly) `$table_obj->name` returns the name of this table.

struct_type (public, readonly) `$table_obj->struct_type` returns the structural type of this table. Current values of `struct_type` can be mandatory, recommended, and optional.

columns/cols (public, readonly) `$table_obj->cols` returns the columns of this table. `$table_obj->columns` is just an alias.

primaryKey (public, readonly) primaryKey() needs one argument, which is either 'ref_col', 'view', 'where', or 'concat'.

```
$table_obj->primaryKey('ref_col')
```

returns the reference column to which this primary key in the table refers to.

```
$table_obj->primaryKey('concat')
```

returns the external columns, that build the concatenated primary key. The old syntax of \$table_obj->primaryKey('ext_cols') is still supported but deprecated.

```
$table_obj->primaryKey('view')
```

returns the viewname of the view, that finally provides the foreignkey through the where clause:

```
$table_obj->primaryKey('where')
```

Often the where clause is 'closing_dt is NULL'. The resulting view then shows only records, which are not closed.

\$table_obj->[sequence | sequences | index | indices | indexes] (public, readonly) They return the index and the sequence entries for the table, either as an array or as an array reference. There are only two methods, the others act like aliases.

usage:

```
my @indices = $table_obj->indices;
my $sequences_ref = $table_obj->sequences;
```

\$table_obj->triggers(\$triggertype) (public, readonly) The method **triggers** takes the following triggertypes as argument:

```
$table_obj->triggers( 'preinsert' );
$table_obj->triggers( 'postinsert' );
$table_obj->triggers( 'preupdate' );
$table_obj->triggers( 'postupdate' );
$table_obj->triggers( 'predelete' );
$table_obj->triggers( 'postdelete' );
```

and returns the triggers for this type. Depending on the calling context they will be returned as a list or as an array reference.

\$table_obj->[datatype | length | default | description | check | modify | foreignkey | label] (public, readonly) Although these methods are column methods, they are kept here for compatibility reasons.

The old, still valid (but deprecated) syntax

```
my $descr = $table_obj->description( $column_name );
```

should now be better written as:

```
my $descr = $column_obj->description;
```

or

```
my $descr = $table_obj->column( $column_name )->description;
```

Apiis::Model::ColumnObj – internal package to provide a column object with methods to access a single column of a table

SYNOPSIS

```
$col_obj = $table_obj->column( $column_name );
```

DESCRIPTION

METHODS \$column_obj->[datatype | length | default | description | check | modify | struct_type | label] (public, readonly)

Example: my \$datatype = \$column_obj->datatype;

The according values from the model file are returned. All these methods are readonly.

check() returns the rules for the current check level. If a check level for a column is defined/exists, this one is taken.

If there is no CHECKn defined for check level n the default CHECK is taken. This also applies if e.g. CHECK2 is defined in the model file but no CHECK1. In this case the default CHECK is taken for CHECK1 as this is undef.

foreignkey (public, readonly)

```
my ($fk_table, $fk_column) = $column_obj->foreignkey;
```

foreignkey() returns the defined foreign key table and the foreign key column for this column, either as an array or as a reference to an array, depending on the callers context.

It returns undef if no foreign key is defined.

6.4.8 Apiis::Errors – Provide error objects for generic error handling in APIIS

SYNOPSIS

```
my $err_obj = Apiis::Errors->new(
    type      => 'CONFIG',
    severity  => 'INFO',
    from      => 'test.Errors',
    msg_short => "No date format defined",
);
```

`Apiis::Errors->new()` creates an error object, that describes an error comprehensively to enable further adequate processing.

DESCRIPTION

`Apiis::Errors` provides an error object with the following traits:

- Error **type**, currently:
 - **DATA** the passed data is not ok (usually in `CheckRules`)
 - **DB** errors from the database (e.g. unique index violation)
 - **OS** errors from the operation system (e.g. full hard disk)
 - **AUTH** errors concerning access rights
 - **PARSE** errors in `ParsePseudoSQL` with parsing pseudo SQL code
 - **CODE** programming errors, e.g. from applications like load objects
 - **PARAM** passed parameter is wrong or missing
 - **CONFIG** one of the configuration files is wrong or has missing entries
 - **INSTALL** there is an error in the `Apiis/Perl` installation
 - **UNKNOWN** is unknown.
- Error **severity**, currently **DEBUG INFO NOTICE WARNING ERR CRIT ALERT EMERG**. These severity values are the same as the unix syslog priorities. See also 'man syslog.conf' under Unix/Linux.
 - **DEBUG** debugging messages for bug hunting
 - **INFO** informational notice
 - **NOTICE** more than information, somebody should notice it
 - **WARNING** influences further processing but is not so severe
 - **WARN** deprecated, use **WARNING**
 - **ERR** error, handled in the normal flow control
 - **ERROR** deprecated, use **ERR**
 - **CRIT** critical error, but can be handled under certain circumstances
 - **ALERT** alarm, immediate intervention necessary
 - **EMERG** no further processing possible (e.g. disk full)
 - **PANIC** deprecated, use **EMERG**
- Error **action**, currently:
 - **INSERT** the error occurred during a database insert
 - **UPDATE** the error occurred during a database update
 - **DELETE** the error occurred during a database delete
 - **SELECT** the error occurred during a database select
 - **FETCH** like **SELECT**
 - **DECODE** the error occurred during an attempt to decode the data

- **ENCODE** the error occurred during an attempt to encode the data
- **UNKNOWN** the action is unknown

The internal structure provides the following fields to describe a certain error:

```
%struct = (
    type          => undef,    # predefined values above
    id            => undef,    # error id
    severity      => undef,    # predefined values above
    action        => undef,    # predefined values above
    from          => undef,    # location where this error comes from
                                # (e.g. sub, rule)
    record_id     => undef,    # id of this record, e.g. record_seq
                                # from inspool
    unit          => undef,    # unit that provides this data
    db_table      => undef,    # database table concerned
    db_column     => undef,    # database column concerned
    data          => undef,    # just handled incorrect data
    ext_fields    => undef,    # involved external fields (array)
    ext_fields_idx => undef,    # index of these external fields (for tabulars)
    ds            => undef,    # data stream name
    err_code      => undef,    # coded error message
    msg_short     => undef,    # main error message for end users
    msg_long      => undef,    # detailed error message
    misc1         => undef,    # user defined scalar
    misc2         => undef,    # user defined scalar
    misc_arr1     => undef,    # user defined array
    misc_arr2     => undef,    # user defined array
    backtrace     => undef,    # backtrace in Carp::longmess style
);
```

Public and internal methods are:

INTERNAL METHODS

new (mostly internal) new creates the object and checks access rights to the object structure.

\$error_obj->[type_values | severity_values | action_values] (all external) These public methods provide read only access to the preconfigured values.

\$error_obj->[from | line | backtrace | record_id | unit | db_table | db_column | data | ext_fields | ext_fields_idx | ds | err_code | msg_short | msg_long | misc1 | misc2 | misc_arr1 | misc_arr2] (all external) These public methods provide read/write access to the structure elements.

print (external) Print the defined elements of this error object in the order of the hash %struct (actually the @struct array). This is mainly used for debugging.

Second input parameter can be a hash with the key:

- **filehandle** – the output then goes to this filehandle instead of STDOUT (default) note: the filehandle has to be passed as a typeglob

Example:

```
$err_obj->print(
    filehandle => *ERR_FILE,
);
```

sprint (external) Return the formatted error message as a string (used by **print**).

sprint_html (external) Return the formatted error message as a string (used by **print**).

syslog_print (external) The error message is formatted for unix syslog (used by \$apiis->log).

6.4.9 Apiis::Misc – Provides some usefull subroutines, mainly for compatibility reasons

SYNOPSIS

```
use Apiis::Misc qw( <subroutine_name> );
```

DESCRIPTION

Apiis::Misc gives you access to the subroutines (not object methods!):

```
show_progress mychomp elapsed is_true
Info Error
LocalToRawDate RawToLocalDate Decode_Date_NativeRDBMS
find_pod_path file2variable mimetype_of
```

You can load some of them by writing:

```
use Apiis::Misc qw( show_progress mychomp );
```

They are also grouped:

```
use Apiis::Misc qw( :Tk );      # exports the Tk routines Info and Error
use Apiis::Misc qw( :date );    # exports the date routines
use Apiis::Misc qw( :all );     # exports all routines
```


Subroutines

show_progress **show_progress** gives you some kind of progress view. It prints a dot every \$mod times (default 100), every \$mod*10 times it prints the number. input: 1.) a reference to the counter (usually starting with 1). 2.) optional: modulus operator \$mod return: none. The counter has to be incremented outside this routine!

MaskForLatex Creates Escape-sequences for print in latex

mychomp By: Chris Nandor (from the Perl Function Repository) removes end-of-line regardless of originating platform of file

Info show an Tk-Info window

```
usage: Info("Infomessage");
```

Error show an error window

```
usage: Error("Errormessage");
```

LocalToRawDate Date conversion.

Standard SQL format seems to be 'DD-MON-YYYY', e.g. '24-MAY-2000'. At least PostgreSQL and Oracle6 accept this. As long as DBI/DBD does not convert different date formats to the standard formats of the databases we have to provide this conversion in apiis. Date::Calc has date formats EU (european format day-month-year) and US (US american format month-day-year) input: type of local dateformat [EU|US] local date return: old version (before Aug. 2001): date in native database format or -1 in case of errors LocalToRawDate('EU', '24.5.2000') will return '24-MAY-2000'. new version: list of ("new_date_string", \$status, \$err_msg) LocalToRawDate('EU', '24.5.2000') will return ('24-MAY-2000',0,undef).

Note: This is old stuff and remains here only for compatibility reasons. It will be removed in the near future. (2005-02-28 heli)

RawToLocalDate RawToLocalDate - change native database date format to EU/US-format (Date::Calc) input: type of local dateformat [EU|US] native database date return: date in local format or -1 in case of errors

RawToLocalDate('EU', '24-MAY-2000') will return '24.5.2000'.

Note: This is old stuff and remains here only for compatibility reasons. It will be removed in the near future. (2005-02-28 heli)

Decode_Date_NativeRDBMS This sub is used like Decode_Date_EU and Decode_Date_US, where 'EU' and 'US' is set in apiisrc. For check_integrity only the native date values from the database is used. It takes DATESEP and DATEORDER from the definitions in Database.pm and parsed the passed date (which is in DATEORDER). It returns (\$year, \$month, \$day) like the Decode_Date_ routines from Date::Calc.

Example: (\$year, \$month, \$day) = Decode_Date_NativeRDBMS('1999-5-13')

Note: This is old stuff and remains here only for compatibility reasons. It will be removed in the near future. (2005-02-28 heli)

elapsed do some profiling: input: array reference with entries in order of Now() output: String "Hours:Minutes:Seconds" elapsed since passed start time

is_true is_true tests the passed scalar value, if it is true or false in the boolean sense.

False are:

- * undef
- * 0 (zero, either as number or as string of length 1)
- * '' (the empty string)
- * all other strings and numbers that are not true

True are:

- * all numbers different from 0
- * the number 0E0 is zero, but true (Perl internal)
- * all strings, which are defined as representing true like 'true', 'yes'. For different languages you can add the according string for 'yes' like 'ja' in german. Only the first character of the string in lowercase is checked.

input: any scalar value that has to be checked for being true or not

return: 1 if the passed value is true, 0 or undef or the empty string otherwise

find_pod_path find_pod_path tries to find the appropriate Perl POD documentation for the invoked programm. It first looks for language specific pod-files (taking \$apiis->language) and continues the search to the less specific versions. Last ressort is the program itself:

```
<programname>_<lang>.pod
<programname>.<lang>.pod
<programname>.pod
<programname>
```

The <programname> contains the complete path, found by the Perl core module FindBin.

file2variable file2variable loads file from the local file system and tries to guess its mime-type

input: file name

return: scalar with the file content and scalar with the mime-type

mimetype_of tries to find out the mimetype of a given filename. It first uses `MIME::Types`, which seems to give the best results. Only if this module is not installed or doesn't find any entry, a second test with `File::Type` is done. **mimetype_of** returns the found mimetype (e.g. `application/vnd.ms-powerpoint`) or `undef` otherwise.

Input: filename with full path.
Return: mimetype or undef

Example:

```
use Apiis::Misc qw( mimetype_of );
my $mt = mimetype_of($file_name);
```

6.4.10 Apiis::CheckFile – Find configuration files

SYNOPSIS

```
my $a = CheckFile->new( file=>'apiis.model' );
```

`CheckFile` gets a file name as argument and looks where it finds this file under `$apiis_local`.

DESCRIPTION

`CheckFile` gets a file name as argument and looks where to find this file under `$apiis_local`. The file name argument can either be

a complete path (e.g. `$apiis_local/model/apiis.model`) or only the file (e.g. `apiis.model`) or only the basename of the file (e.g. `apiis`)

Examples:

```
my $a = CheckFile->new(file=>'apiis.model');
my $a = CheckFile->new(file=>'apiis');
my $a = CheckFile->new(file=>"$apiis_local/model/forms/abc");
my $a = CheckFile->new(file=>"../..apiis/reports/jjj");
```

The recognized filename extensions are defined in `$self->{_suffixes}`, the locations for doing the search in `$self->{_locations}`.

6.5 Authentication and Authorisation

6.5.1 DESCRIPTION

`Auth.pm` is method of record object and contains subroutines needed for the authentication process.

6.5.2 SUBROUTINES

`_auth`

This subroutine is responsible for whole authentication process.

1. Get access rights from the database

2. Dependant from sql action type

- (a) if DELETE then only recreates list of all clases on which user can executed delete operations. This list is returned by "get_ar" and is recreated to the hash structure.
- (b) if INSERT or UPDATE then checks access rights for the columns defined in this statement

check_ar

Subroutine checks access rights for the columns. Algorithm compares column names from the sql statement to column names defined in the access hash and returns hash with clases in which user can executed this sql statement. Class is added to this hash only if user have defined access rights for all columns (from sql statement) in this class.

Returned hash for expand the where clause:

```
@ext =(
    (
        COLUMN    => "class",
        OPERATOR  => "=",
        VALUE     => PL,
    ),
    (
        COLUMN    => "class",
        OPERATOR  => "=",
        VALUE     => DE,
    )
)
```

redo_clases

6.5.3 AUTHORS

Marek Imialek <marek@tzv.fal.de>

6.5.4 Apiis::Auth::Role.pm

SYNOPSIS

```
$role = Apiis::Auth::Role->new( role_shortcut => $role_shortcut, );
```

DESCRIPTION

This is a module for creating an object for handling user roles. These object are used in the authentication process. All information about roles are taken from the Roles.conf file.

METHODS

role_shortcut returns the main role name which is defined in Roles.conf file

short_name returns short role name

long_name returns long role name

description returns description of role

policies returns policy numbers; policies number are taken from the role subsectione

role_type returns role type

role_id get next sequence value

db_policy get policy for the database from db_policy section

os_policy get policy for the operating system from os_policy section

AUTHORS

Marek Imialek <marek@tzv.fal.de>

6.5.5 Apiis::Auth::AppAuth – object for provading data about user access rights for the applications

SYNOPSIS

This object is used to check user access right for the applications which ara curently defined in the database.

DESCRIPTION

Object is created by the one of Apiis object method (\$apiis->join_auth('user_login')). This creates Auth object for the user which is curently log-in and join it to the \$apiis structure.

METHODS

new (public)

returns an object reference for a new Auth object.

__get_user_roles (internal)

retrieves all role_id from table 'roles' for current user

__get_user_id (internal)

retrieves current user id

`_get_policy_ids` (internal)

retrieves all policies for current role

`print_os_actions` (public)

prints all applications or actions with their classes which user can execut

example: `$apiis->Auth->print_os_actions;`

`os_actions`

method return list of all actions which are allowed for the user (if you run it with
If you run it with the parameter "action type" then you can get the list of allowed
You can use following action type: program, form, rapor,t subroutine, www, action.
in the AccessControl.pm

example: `$apiis->Auth->os_actions`
`$apiis->Auth->os_actions('program')`

`types_of_actions` (public)

returns all type of actions which are curently allowed for the user.

example: `$apiis->Auth->types_of_actions`

`check_os_action` (public)

check that user can executs action (action name is defined as a parameter).

example: `$apiis->Auth->check_os_action('runall_ar.pl','program');`

AUTHORS

Marek Imialek <marek@tzv.fal.de>

6.5.6 Apiis::Auth::AccessControl – used by the runall.pl and access_control.pl scripts to define user access rights

SYNOPSIS

Adding, deleting roles and users in the Apiis system.

DESCRIPTION

These subroutines are used to define access rights in the system. New roles and users are created on the basis of information which are defined in the `$APIIS_LOCAL/etc/Roles.conf`. Roles and users name are set as a parameters.

SUBROUTINES**access_rights**

this subroutine can be used to define access rights directly in the code without access_control

creates_schema

this subroutine creates individual user schema

creates_user

this subroutine adds new user.

creates_role

this subroutine adds new role.

assigns_role

this subroutine assigns role to the user.

creates_db_policies

This subroutine reads database policies from the Roles.conf file and adds it to the database. Only these policies are added which are defined for the current role.

assigns_db_policies

this subroutine assigns database policies to the current role.

creates_os_policies

This subroutine reads os policies from the Roles.conf file and adds it to the database. Only these policies are added which are defined for the current role

assigns_os_policies

this subroutine assigns operating system policies to the current role.

creates_access_view

this subroutine creates user access view.

check_policies

This subroutine checks that policies which we want to load already are defined in the database.

del_user

this subroutine deletes user from the system.

del_role

this subroutine deletes role from the system.

del_role_from_user

this subroutine revoke role from the user.

revoke_priv

this subroutine revoke user privileges from the PostgreSQL.

select_db

this subroutine print information about users and roles which are curently defined

public_views

this subroutine creates system of the public views in the user schema.

col_in_classes

This subroutine querys DB about allowed columns for current user. Query is executed with select action and current table name as a paramaters. Returned list of columns is added to the hash with class names.

main_columns

This subroutine creates main list of column for the current table . Only these columns are taken to the list, to which user have access rights (in several classes). If there is a translation table then this list is created from two tables. This list is needed to create structure of view via UNION (we have to have list of all columns which will be in the view).

class_columns

Compares main column list to the column allowed for the user in several classes. Columns list is created for each class. Columns order for the user have to be the same like in main list. Value NULL is putted to the list if some column occurs on the main list and user haven't access rights to this column in this class. This all is needed to create final sql where "union all" expression is used.

creates_view

This subroutine creates user view for current table.

drop_views

this subroutine drop current view from user schema.

query_db

this subroutine executes system SQL statements.

change_password

this subroutine is used to change user password.

creates_views_v_ this code was copied from MakeSQL.pm module and partially changed. This subroutine is used to create "v_" view under user schema.

Note (2008-04-08 heli): As MakeSQL changed to allow self-referencing foreign keys I also had to change this part of it. :^(

AUTHOR

Marek Imialek <marek@tzv.fal.de>

6.6 The Database

6.6.1 Apiis::DataBase::Init – Basic database initialisation

SYNOPSIS

Database Initialisation based on the configuration in apiisrc and model file

DESCRIPTION

When loading the model file, database initialisation usually is done automatically. For certain special cases it can be delayed.

\$apiis->DataBase->disconnect() **\$apiis->DataBase->disconnect()** disconnects the global database handle \$dbh (**\$apiis->DataBase->dbh**) from the database. If you pass a handle to **disconnect()** like:

```
$apiis->DataBase->disconnect( $my_db_handle );
```

then this \$my_db_handle will be disconnected.

commit(), user_commit(), sys_commit(), rollback(), user_rollback(), sys_rollback() These methods commit/rollback all transactions for either the system or the user database handle.

While

```
$apiis->DataBase->sys_commit;  
$apiis->DataBase->sys_rollback;
```

and

```
$apiis->DataBase->commit;  
$apiis->DataBase->rollback;
```

commit/rollback changes to the system database handle,

```
$apiis->DataBase->user_commit;  
$apiis->DataBase->user_rollback;
```

perform this for the user database handle.

If the commit/rollback fails, an error object is created and an error status of 1 is returned and the \$object->status is set to 1.

seq_next_val

```
$apiis->DataBase->seq_next_val( <sequence_name> );
```

returns the next value of the sequence <sequence_name>.

_get_users (internal) **_get_users** retrieves all configured users from table 'users' and fills an internal datastructure to access the needed values.

_get_users is invoked by `Apiis::Init::_join_database`.

Note: As this served as some kind of caching and we have performance problems during initialization (especially for the web interface), this approach will be deactivated and replaced by queries for every single user (based on the login data). (20.12.04 - heli)

_get_user (internal) **_get_user** retrieves the data for the passed user from table 'users' and fills an internal datastructure to access the needed values.

_get_user is invoked by `Apiis::DataBase::Init::verify_user`.

_get_user_roles (internal) All roles of the `$apiis->DataBase->users` are fetched from database and stored in the user objects.

_get_user_roles is invoked by `Apiis::Init::_join_database`.

users (public) **users** returns all database users of this project.

user (public) **user** returns a User object for the passed user.

verify_user (internal) **verify_user** checks the login data of the passed user object (name and password) against the internal values from the database.

crosstab **crosstab** is a wrapper around the CPAN-Modules `DBIx::SQLCrosstab` and `DBIx::SQLCrosstab::Format`. They allow a convenient way of creating cross tabulations from the database and outputting it into different formats. See 'man `DBIx::SQLCrosstab`' and 'man `DBIx::SQLCrosstab::Format`' for details.

`$apiis->DataBase->crosstab` integrates `DBIx::SQLCrosstab` into the apiis framework. It assumes all necessary parameters getting provided with via a hash reference.

Input parameters:

```
* $hash_ref->{params} with all parameters according to DBIx::SQLCrosstab.
  This includes a database handle (either user_dbh or sys_dbh).
* $hash_ref->{format} defines the output format
* $hash_ref->{aux} passes an additional parameter, which some formats
  expect, e.g. as_xls('filename'):
    $hash_ref->{format} = 'as_xls';
    $hash_ref->{aux}    = 'filename';
```

Output parameters:

```
* According to the documentation of DBIx::SQLCrosstab::Format
```

Example:

```
my $return_val = $apiis->DataBase->crosstab($hash_ref);
```

6.6.2 DBCreation.pm

DESCRIPTION

This module is used in runall process to create initial database structure. The structure of the database is taken from the model file. New database is created in the system user schema (system user name is taken from the modelfile) and the public schema is removed. Then the languages are loaded from defined file and information about node is inserted (from apissrc settings). At the end sequences are correctly set.

SUBROUTINES

CreateDatabase This subroutine call all subroutines which are written in this file. The parameters are defined as: - \$db_encoding - character encoding which will be used in the database - \$db_name - database name - \$user_creator - user name which creates database - \$lang_file - file name where the initial languages are written (language.dat in the reference database) - \$lang_dir - directory for the language file

InitDB This subroutine creates initial database structure. PUBLIC schema is removed and the database is placed in the system user schema. System user schema name is taken from the user name defined in the modelfile.

LoadLang This subroutine load initial languages to the languages table.

LoadNodeData This subroutine load initial information about node. These information are inserted in the nodes table.

SetSequences This subroutine sets initial sequences.

AUTHORS

Marek Imialek <marek@tzv.fal.de>

6.6.3 Apiis::DataBase::User – collecting and providing user data

SYNOPSIS

```
my $usr_obj = Apiis::DataBase::User->new( id => <userid>, %args );
```

DESCRIPTION

To create a new User object, we need at least a user id. A User object is created and returned.

```
my $usr_obj = Apiis::DataBase::User->new( id => <userid> );
```

Other parameters can be passed to fill the object at creation time:

```
my $usr_obj = Apiis::DataBase::User->new(
    id      => <userid>,
    password => <top_secret>,
);
```

When you run

```
$apiis->join_user( user_obj => $this_obj, %args );
```

the User object `$this_obj` is joined into the `$apiis` structure. This can happen only once as only one user can run a program at a time. If no `user_obj` is passed, `join_user` falls back to ask for login data by itself.

Nevertheless it is possible, to create other User object, e.g. to insert a new user into the database.

With

```
my $user_obj = $apiis->DataBase->user( <userid> );
```

you can retrieve all user information, stored in the database, into a User object.

TODO

Maybe we get some class `Apiis::Person` later, whereof `Apiis::DataBase::User` is a subclass. So User should be restricted to the data which is needed for database connection (authentication and authorisation), whilst other personal data like name, address, etc. should be retrieved from the unit/naming/address setup via some foreign key. In table users, column login will then become the primary key (enforce uniqueness) and `user_id` could point to unit (or somewhere else).

METHODS

new (public) `new()` returns an object reference for a new User object.

roles (public) `$usr_obj->roles` returns the roles of this user either as an array or an arrayreference. Roles are stored here by:

```
$usr_obj->roles( \@these_roles );
or
$usr_obj->roles( 'role1', 'role2', 'roleN' );
```

password

```
$usr_obj->password
```

returns the encrypted password. If a new password is passed with

```
$usr_obj->password( <new_password> );
```

this password is first encrypted and then stored. If you want to store an already encrypted password (like one from the database) you have to invoke it

```
$usr_obj->password( <new_password>, encrypted => 1 );
```

authenticated (public) This is a flag to show successful authentication of this user against his database password.

It is read-only to allow setting of this flag only from inside of **verify_user**

print (external) Return the contents of the User object as a string, nicely formatted. The elements of this object are printed according to the order of the @methods array, if they contain data.

print (external) Print this user object to STDOUT (default) or the passed filehandle with the formatting of sprint.

Examples: \$usr_obj->print; \$usr_obj->print(filehandle => *FILE);

user_language_id (external) \$usr_obj->user_language_id returns the language id as stored in table ar_users (new Auth/AR setup).

lang_id (external) \$usr_obj->lang_id returns the language id as stored in table users (old, deprecated Auth/AR setup).

language (external) \$usr_obj->language returns the value of iso_lang from table languages with the given lang_id. It is preset with \$apiis->language as default.

6.6.4 The Database Record Object

6.6.4.1 Apiis::DataBase::Record – package for DataBase Record objects

SYNOPSIS This base package provides the functionality and methods needed for database record object types.

DESCRIPTION The public and internal methods of this base class are described below.

METHODS

new (public) **new()** returns an object reference for a new record object.

name (public, readonly) Returns the name of this record (usually identical with the tablename of the database record):

columns (public, readonly) Returns an array of all column names of this record in the order of the model file structure.

column (public, readonly) Returns a column object by column name.
Example:

```
my $col_obj = $record_obj->column($thiscolumn);
```

addcolumn (public) Adds a column object to the record. Maybe only for internal use.

delcolumn (public) Deletes a column object from the record. Maybe only for internal use.

rows | value | fk_table | values (public, read/write) Methods for return status parameters, e.g. for SQL query results.

print print prints out the defined column values of the Record object, both internal and external values and the associated ext_fields.

Example:

```
$record_obj->print;
```

There are some switches to control the behaviour of **print**:

```
quiet => 1          # if set to 1, informational output is reduced
columns => \@cols   # prints out only the defined columns
m_int  => 1          # displays also mirrored internal data
m_ext  => 1          # displays also mirrored external data
id_set => 1          # displays also the id_set definition of the column
all    => 1          # includes m_int, m_ext, id_set
sprintf => 1        # doesn't print to STDOUT but returns an arrayref to the
                    # output lines
```

Example:

```
$record_obj->print(
  { quiet    => 1,
    columns => [qw/ db_animal guid /],
    m_int   => 1,
    id_set  => 1,
  }
);

$record_obj->print(
  { quiet    => 1,
    columns => [qw/ db_animal guid /],
    all     => 1,
  }
);

my $output_ref = $record_obj->print( { sprintf => 1 } );
print join("\n", @$output_ref);
```

The parameters have to be passed as a hash reference, the columns as an array reference.

6.6.4.2 Apiis::DataBase::Record::Column – package for DataBase Record columns

SYNOPSIS **Apiis::DataBase::Column** creates database columns that build an **Apiis::DataBase::Record** and provides methods to access them.

DESCRIPTION The public and internal methods of this class are described below.

METHODS

use_entry_view (public) **use_entry_view** is a column method to allow encoding distinguish between values from a table <Table> or from the view entry_<Table>. In case of transfer this flag decides, if encoding takes the db_animal from entry_transfer (an active animal) or from transfer directly, which may be any db_animal with these external data.

id_set (public) The read/write method **id_set** stores and offers the ID set information to retrieve the right record from table transfer when decoding db_animal.

Input:

```
$rec_obj->column('db_animal')->id_set('HB');           # single value
$rec_obj->column('db_animal')->id_set('HB', 'Piglet'); # list
my @id_sets = (qw/ HB Piglet Lifetime /);
$rec_obj->column('db_animal')->id_set(@id_sets);       # array
$rec_obj->column('db_animal')->id_set(\@id_sets);      # arrayref
```

Output:

```
my $id_set_ref = $rec_obj->column('db_animal')->id_set # arrayref
my @id_sets    = $rec_obj->column('db_animal')->id_set # array
```

best_id_set (public) The read/write method **best_id_set** stores that ID set information, that has been the best/first while trying to decode through the array of ID sets in **id_set()**.

6.6.4.3 Fetch

SYNOPSIS

```
# fill record with query data, then:
$record_obj->fetch(
    expect_rows    => 'one',           # ['one'|'many']
    expect_columns => qw/ db_sex db_breed/, # all columns if absent
    order_by       => [                # optional sorting
        { column => 'db_sex',  order => 'desc' },
        { column => 'db_breed', order => 'asc' },
    ],
    user => 'system',                 # optional for internal use
);
```

`Apiis::DataBase::Record::Fetch` fetches the specified record(s), creates a record object for each of them and returns an array of these objects.

DESCRIPTION The query is created according to the already filled-in data in this record object.

All columns of this record object, which contain data, build 'column=value'-pairs. The operator is currently only '=' and all data fields are ANDed.

Special allowed values are 'null' and 'not null', which are reflected in the resulting where clause. Both strings are case insensitive.

Example:

```
$rec_obj->column('exit_dt')->extdata('not null');
```

or

```
$rec_obj->column('db_breeder')->intdata('null');
$rec_obj->column('db_breeder')->encoded(1);
```

The qualifiers `expect_rows()` and `expect_columns()` can either be provided as methods to the record object or as hash parameters to fetch.

`expect_rows()` can have the values 'one' and 'many'(default).

If `expect_columns()` is omitted, all columns of the record are retrieved. Every records also retrieves the rowid/oid.

Example:

```
my $rec_obj = Apiis::DataBase::Record->new( tablename => 'codes' );
$rec_obj->column('class')->extdata('BREED');
my @query_objects = $rec_obj->fetch(
    expect_rows      => 'many',
    expect_columns => [qw/ db_code ext_code /],
    sort_by          => [ { column => 'ext_code', order => 'asc' } ],
);
```

This query returns columns `db_code` and `ext_code` of all the records from codes, where the BREED is coded. They are sorted by `ext_code` in ascending order. The resulting rows are packed into separate record objects each and passed back in an array of record objects.

`$rec_obj->expect_rows('many')` is default and can be omitted.

Another example:

```
my $rec_obj = Apiis::DataBase::Record->new( tablename => 'animal' );

# if you know the internal code:
$rec_obj->column('db_animal')->intdata(8608);
# you must set the encoded(1) flag for internal data:
$rec_obj->column('db_animal')->encoded(1);

# or, if you have the external data:
$rec_obj->column('db_animal')->extdata( 'society|sex', '32|1', '63' );
```



```
# now specify the query:
$rec_obj->expect_columns(qw/ db_sex db_breed /);
$rec_obj->expect_rows('one');
$rec_obj->fetch;
```

If you provide the external data, it is encoded before the query is created:

```
SELECT db_sex,db_breed from animal where db_animal = 8608;
```

If this results in more than one record (which should not happen), an error is risen.

Note: If you don't want the record to get encoded (perhaps you provide internal data anyway) you have to set `$record->encoded(1)`, which will skip encoding. In case of providing internal data without external one, leaving out `$record->encoded(1)` will yield an error. During encoding, the internal value will simply get deleted.

Ordering the query result can be done with:

```
$record_obj->fetch(
  order_by => [
    { column => 'birth_dt', order => 'desc' },
    { column => 'parity',   order => 'asc'  },
  ],
);
```

`order_by` expects an array reference of hash references. According to the SQL standard, this query is sorted by column `birth_dt` in descending order and, for equal values in `birth_dt`, by column `parity` in ascending order.

This makes mainly sense for columns without encoded values as the sorting happens on the internal database values.

6.6.5 Modify Rules

6.6.5.1 CommaToDot

SYNOPSIS `CommaToDot` substitutes commas with dots.

DESCRIPTION In some countries, e.g. Germany, often the comma is used as the decimal separator of a number. The method `CommaToDot` simply changes the comma to dot to satisfy the database. `CommaToDot()` is usually used as a `MODIFY`-rule in the model file.

6.6.5.2 ConvertBool

SYNOPSIS `ConvertBool` converts `YyJjNn` to the boolean values 1 or 0.

DESCRIPTION `ConvertBool` is usually used as a `MODIFY`-rule in the model file.

6.6.5.3 DotToColon

SYNOPSIS `DotToColon` substitutes dots with colons.

DESCRIPTION **DotToColon** is useful for fast typing of date/time values on numerical keyboards.

Example: 16.34.00 -> 16:34:00

DotToColon() is usually used as a MODIFY-rule in the model file.

6.6.5.4 LowerCase

SYNOPSIS **LowerCase()** converts the data to lower case

DESCRIPTION It is usually used as a MODIFY-rule in the model file.

AUTHORS Helmut Lichtenberg <heli@tzv.fal.de>

6.6.5.5 SetNow

SYNOPSIS **SetNow()** returns the current date and time.

DESCRIPTION **SetNow()** returns the current date and time. It is usually used as a MODIFY-rule in the model file. If you need the current time outside the model file, better use `$apiis->now()`.

6.6.5.6 SetUser

SYNOPSIS **SetUser()** returns the current user name.

DESCRIPTION **SetUser()** sets the passed column extdata to the current user name. It is usually used as a MODIFY-rule in the model file.

6.6.5.7 UpperCase

SYNOPSIS **UpperCase()** converts the data to upper case

DESCRIPTION It is usually used as a MODIFY-rule in the model file.

AUTHORS Helmut Lichtenberg <heli@tzv.fal.de>

6.6.6 Check Rules

6.6.6.1 DateDiff

SYNOPSIS Syntax in model file:

```
DateDiff min_diff max_diff compare_date [reference_column]
```

Examples: CHECK => ['DateDiff 0 365 2001-03-22'], CHECK => ['DateDiff 1 100 buy_dt'], CHECK => ['DateDiff 1 50 animal=>birth_dt db_animal'],

DESCRIPTION DateDiff takes the current value (\$data) of the passed column and computes the difference to the date, given in the third parameter (compare_date). If the difference (in days) between \$data and compare_date is in the range given by min_diff and max_diff, DateDiff will return 0 for success, otherwise 1. In other words:

```
min_diff <= ($data - compare_date) <= max_diff    # success
```

compare_date can either be a fixed format date like '2001-03-22' (must be in ISO 8601 format) or a date in a column of this record or a date in the column of another table. In the latter case, the format is 'tablename=>columnname' and you additionally have to give the referencing column of both tables. This referencing column connects both tables (usually a foreign key).

Examples:

```
# fixed format:
DateDiff 1 365 2000-01-15

# compare to another column in the current record:
DateDiff 30 50 buy_dt

# compare to another table.column:
DateDiff 80 120 animal=>birthdate db_animal
```

AUTHORS Helmut Lichtenberg <heli@tzv.fal.de>

6.6.6.2 ForeignKey

SYNOPSIS Syntax in model file:

```
ForeignKey fk_table fk_column [column=value]
```

DESCRIPTION The internal data of the current table and column must have an according entry in fk_table.fk_column.

Undefined data (NULL) does not violate the rule. It must be checked with NotNull.

ForeignKey() returns 0 in case of success, otherwise it creates a descriptive record error object and returns 1;

For internal en-/decoding, the ForeignKey rule is somehow violated with additional parameters, which are not needed for the pure FK-checking. If we have a FK-definition:

```
ForeignKey codes db_code class=BREED
```

the FK-checking only looks in table codes, column db_code. The additional 'class=BREED' entry is used for the coding stuff.

check_ForeignKey() **check_ForeignKey()** checks the correctness of the input parameters.

In case of errors it sets \$self->status and additionally returns a non-true returnvalue.

AUTHORS Helmut Lichtenberg <heli@tzv.fal.de>

6.6.6.3 IsAFloat

SYNOPSIS The passed data must be a floating point number

DESCRIPTION

IsAFloat() The value of the current column has to be a floating point number. Empty values are allowed.

Returnvalues: nothing in case of success local status with true value, errors are stored in \$record->errors

check_IsAFloat() **check_IsAFloat()** checks the correctness of the input parameters.

In case of errors it sets \$self->status and additionally returns a non-true returnvalue.

Checks are: Existence of additional parameters

AUTHORS Helmut Lichtenberg <heli@tzv.fal.de>

IsAFloat The passed value \$data is checked to be a float. The , is not substituted to decimal point as this subroutine only return 0 or 1, not the (substituted) value. This is a job for the Modify rule CommaToDot.

Returnvalues: 0 if \$data is a legal float, 1 if \$data contains illegal chars

6.6.6.4 IsANumber

SYNOPSIS Checks, if the provided data is a number.

DESCRIPTION

IsANumber() The value of the current column has to be a number. Empty values are allowed. The test is done by comparing firstly \$value with \$value+0 and, if this fails, with a more complex regex.

Returnvalues: nothing in case of success local status with true value, errors are stored in \$record->errors

check_IsANumber() **check_IsANumber()** checks the correctness of the input parameters.

In case of errors it sets \$self->status and additionally returns a non-true returnvalue.

Checks are: Existence of additional parameters

is_a_number() (internal) **is_a_number()** is an internal routine which is not bind to the record object. Some rules want to check the passed parameter (e.g. Range), if they are numbers. **is_a_number()** gets as input the value, which it has to check. **IsANumber** uses **is_a_number()**, too.

In case of errors it returns a non-true returnvalue.

AUTHORS Helmut Lichtenberg <heli@tzv.fal.de>

6.6.6.5 IsEqual

SYNOPSIS Syntax: IsEqual \$table \$id_column \$column compare_constant [nullok]

DESCRIPTION **IsEqual()** is usually used as a CHECK-rule in the model file.

It checks, if a record, identified by \$table.\$id_column has the value 'compare_constant' in column \$column.

Example: IsEqual animal db_animal db_sex male

This CHECK-rule can be attached to a column like db_sire in service and tests if the animal ID, given in the passed column, points indeed to a male animal. The record from table animal, where db_animal is equal to the data of the current column, must have an entry 'male' in column db_sex. The 'compare_constant' part is a fixed value and specified as external code (codes.ext_code).

Returnvalues:

1. 0 if the retrieved record from \$table.\$id_column has an entry of 'compare_constant' in column \$column.

If the optional parameter 'nullok' is given, an undefined value for this column (or no retrieved record) will also be accepted.

2. All other cases indicate error and an error message exists.

check_IsEqual() **check_IsEqual()** checks the correctness of the input parameters.

In case of errors it puts an error into \$self->errors and additionally returns a non-true returnvalue.

Checks are:

Missing parameters

Last parameter must be 'nullok' if it exists

AUTHORS Helmut Lichtenberg <heli@tzv.fal.de>

VERSION

\$Revision: 1.15 \$

LastAction Syntax:

LastAction table=>chain_col LA_chain min_diff max_diff

LastAction is a conditional DateDiff depending on the value of LA_chain, which is stored in table=>chain_col. The dates to compare are table=>chain_col_dt (_dt extension is hardcoded!) and the passed current value \$data.

If the date difference between `chain_col_dt` and `$data` for this `LastAction` in `table=>chain_col` is **not** within the defined range, the `(error)status $self->status` is set to 1 and an appropriate error object is created. If this rule is not violated, the `(error)status` is 0.

Example for an entry in the model file:

```
CHECK => ['LastAction animal=>la_rep
          SERVICE 18 62
          FARROW  40 80'],
```

Note! As `LastAction` is a very specific check rule there are some details hardcoded. The connecting column between tables (foreign key) is `db_animal` in both tables. The column, that contains the date for the last action is assumed to have the last-action-column name with `'_dt'` appended.

If `LastAction` turns out to be a useful check rule for other purposes where the hardcoding is an obstacle, it can be rewritten in a generic manner, likely with the drawback of some changes in parameter passing.

skip_LastAction() `skip_LastAction()` returns the actions, when checking of this rule should be skipped. For `LastAction`, checks during an update operation are useless.

Input: none Output: arrayref

check_LastAction() `check_LastAction()` checks the correctness of the input parameters.

In case of errors it sets `$self->status` and additionally returns a non-true returnvalue.

6.6.6.6 AUTHORS

Helmut Lichtenberg <heli@tzv.fal.de>

6.6.6.7 List

SYNOPSIS `List()` is the poor man's foreign key. The data is checked against a small list which is provided in the model file.

DESCRIPTION The model file can provide a small list where the data is checked against. Example, column `db_sex`:

```
CHECK => ['List Male Female'],
```

The external data is allowed to have the values 'Male' or 'Female'. The number of List entries is not limited:

```
CHECK => ['List val1 val2 ... valN'],
```

To circumvent upper/lower case problems you can combine `MODIFY` and `CHECK` rules:

```
MODIFY => ['UpperCase'],
CHECK  => ['List MALE FEMALE'],
```

The data is first modified and then checked.

Undefined or NULL data is accepted as it can get controlled with NotNull.

Returnvalues: 0 if \$data is one of the list values (success), 1 otherwise (error)

check_List() **check_List()** checks the correctness of the input parameters.

In case of errors it sets \$self->status and additionally returns a non-true returnvalue.

AUTHORS Helmut Lichtenberg <heli@tzv.fal.de>

6.6.6.8 NoCheck

SYNOPSIS The Rule **NoCheck** is checks nothing. It is intended to be a noop rule to overwrite existing ones on a lower CHECK-level.

DESCRIPTION

NoCheck() Syntax: NoCheck

Returnvalues: 0 if data is within this range, 1 otherwise errors are stored in \$record->errors

A non-true return value can only happen if the rule **NoCheck** is defined incorrectly in the model file, i.e. an additional parameter is provided.

check_NoCheck() **check_NoCheck()** checks the correctness of the input parameters.

In case of errors it returns a non-true returnvalue.

Checks are: existence of additional parameters

AUTHORS Helmut Lichtenberg <heli@tzv.fal.de>

6.6.6.9 NoNumber

SYNOPSIS

DESCRIPTION Checks, if the provided data is not a number.

NoNumber() The value of the current column must not be a number. Empty values are allowed.

Returnvalues: * nothing in case of success * local status with true value in case of failure, errors are stored in \$record->errors

check_NoNumber() **check_NoNumber()** checks the correctness of the input parameters.

In case of errors it sets \$self->status and additionally returns a non-true returnvalue.

Checks are: Existence of additional parameters

AUTHORS Helmut Lichtenberg <heli@tzv.fal.de>

6.6.6.10 NotNull

SYNOPSIS `NotNull()` checks, if the data has a defined value

DESCRIPTION The passed value `$data` is not allowed to be undefined or empty. It may have the numeric value 0.

`NotNull()` is usually used as a CHECK-rule in the model file.

AUTHORS Helmut Lichtenberg <heli@tzv.fal.de>

6.6.6.11 Range

SYNOPSIS

DESCRIPTION The Rule **Range** is given a range of values in the model file. It then checks, if the provided data is within this range.

Range() Syntax: `Range min_value max_value`

Is the data within a range? `min_value` and `max_value` are predefined in the model file.

Returnvalues: 0 if data is within this range, 1 otherwise errors are stored in `$record->errors`

check_Range() **check_Range()** checks the correctness of the input parameters.

In case of errors it sets `$self->status` and additionally returns a non-true returnvalue.

Checks are: if `min_value` and `max_value` are defined if `min_value` and `max_value` are numbers

BUGS **Range** is intended to work only for numerical values.

AUTHORS Helmut Lichtenberg <heli@tzv.fal.de>

6.6.6.12 ReservedStrings

SYNOPSIS **ReservedStrings** checks, if the data contains strings, that are not allowed as they are used e.g. as concatenation symbol.

DESCRIPTION **ReservedStrings** checks if the passed data contains one of the reserved strings which are defined in `apiisrc`. NULL data (undefined or empty) will pass successfully.

ReservedStrings is usually used as a CHECK-rule in the model file.

check_ReservedStrings **check_ReservedStrings** checks the correctness of the input parameters. In case of errors it puts an error into `$record->errors` and returns a non-true returnvalue.

Checks are: Existence of parameters

AUTHORS Helmut Lichtenberg <heli@tzv.fal.de>

6.6.6.13 Unique

SYNOPSIS

DESCRIPTION

Unique()

check_Unique() **check_Unique()** checks the correctness of the input parameters.

In case of errors it sets `$self->status` and additionally returns a non-true returnvalue.

AUTHORS Helmut Lichtenberg <heli@tzv.fal.de>

Unique Syntax:

Unique \$table \$data_column [\$column=\$value]

Unique looks in database if the passed data is unique within this combination of column(s). \$table is the concerning database table. \$data_column is the column of the passed data. It therefore must **not** have the =value. To create composite keys you can specify additional columns with the expected value. Note: char values have to be surrounded by "".

Example:

Unique(employee name department='sale' salary=3000 \$data);

Returnvalues:

1. if \$data does exist more than once in this Unique combination in the table,
2. otherwise, also accepting NULL values.

6.6.7 Triggers

6.6.7.1 SetGuid

SYNOPSIS Example entry in Model file:

```
PREINSERT => ['SetGuid guid'],
```

SetGuid()

check_SetGuid() **check_SetGuid()** checks the correctness of the input parameters. In case of errors it returns a non-true returnvalue.

AUTHORS Zhivko Duchevev <duchev@tzv.fal.de> Helmut Lichtenberg <heli@tzv.fal.de>

6.6.7.2 SetNode

SYNOPSIS Example entry in Model file:

```
PREINSERT => ['SetNode owner'],
```

```
SetNode()
```

check_SetNode() **check_SetNode()** checks the correctness of the input parameters. In case of errors it returns a non-true returnvalue.

AUTHORS Zhivko Duchevev <duchev@tzv.fal.de> Helmut Lichtenberg <heli@tzv.fal.de>

6.6.7.3 SetVersion

SYNOPSIS Example entry in Model file:

```
PREINSERT => ['SetVersion version'],
```

```
SetVersion()
```

check_SetVersion() **check_SetVersion()** checks the correctness of the input parameters. In case of errors it returns a non-true returnvalue.

AUTHORS Zhivko Duchevev <duchev@tzv.fal.de> Helmut Lichtenberg <heli@tzv.fal.de>

6.6.8 Using SQL**6.6.8.1 DataStream**

SYNOPSIS

```
$ds = Apiis::DataBase::SQL::DataStream->new(
                                                    ds      => $ds,
                                                    debug   => $debug
);
```

This is the module for creating an object for handling datastreams.

DESCRIPTION Creates the internal structure for the hash previously used in the batch processing and provides methods for accessing them.

Public and internal methods are:

PUBLIC METHODS

ds - returns the name of the datastream

job_start - sets or returns the time when the job was started

job_end

- sets or returns the time when the job was finished

records_total - sets or returns total number of processed records

records_error - sets or returns the number of errors that occur in the records - it is not precise and can give only impression of real number of errors

records_ok - sets or returns total number of records that was OK

data - sets or returns the data

LO_keys - sets or returns the list of LO_keys

ext_unit - sets or returns the ext_unit that has supplied this record

record_seq - sets or returns the record number from INSPOOL system

target_column - sets or returns an extra information for linking the errors to special column

debug - sets or returns debug level

verbose - sets or returns the verbose mode (more output messages)

sth_update_inspool - sets or returns statement handle for updating inspool

sth_inspool_err - sets or returns statement handle for inserting new record in inspool_err

sth_ds - sets or returns statement handle for reading record from inspool

sth_load_stat - sets or returns statement handle for inserting new record in load_stat

status - sets or returns the object status - inherited from apiis

errors - sets or returns list of error objects - inherited from apiis

_standard_keys (internal) - encapsulates the names of the automatically created methods

_accessible (internal) - checks if the method is read-only or read-write

new - creates the datastream object

`_init (internal)` - initializes the counters and prepares several database statements

`CheckDS` Verifies if the number of elements in the DS is the same as the number of LO keys

`PostHandling` Writes the summary statistics on the screen and in `tableload_stat` and `error_messages` in table `inspool_err`

AUTHORS Helmut Lichtenberg <heli@tzv.fal.de> Zhivko Ducheve <duchev@tzv.fal.de>

6.6.8.2 Apiis::DataBase::SQL::DirectSQL Direct access to SQL

SYNOPSIS

```
$apiis->DataBase->user_sql( $sql_statement );
$apiis->DataBase->sys_sql( $sql_statement );
```

DESCRIPTION Both of these methods point to the internal work horse `_sql` with the invocation parameters as a hash reference:

```
_sql( { statement => $sql_statement, user => 'system' } )
```

in case of `sys_sql` or without the 'user' parameter for `user_sql`.

`_sql` creates its own statement object, which is returned.

METHODS

`user_sql` | `sys_sql` | `_sql` The only input parameter for **`user_sql`** and **`sys_sql`** is a valid SQL statement. Returned will be a statement object with the following methods:

Examples:

```
my $statement_obj    = $apiis->DataBase->user_sql('select * from codes');
my $statement_handle = $statement_obj->handle;
my $processed_rows   = $statement_obj->rows;
if ( $statement_obj->status ) {
    for my $error ( $statement_obj->errors ) {
        # handle error:
        $error->print;
    }
}
```

See 'man DBI' for detailed information about the statement handle and rows.

6.6.8.3 PseudoStatement**SYNOPSIS**

```

    $statement = Apiis::DataBase::SQL::PseudoStatement->new(
        pseudosql      => $sqltext,
        data_hash      => \%data_hash
    );

```

This is the module for creating an object for parsed PseudoSQL.

DESCRIPTION Creates the internal structure for the important elements of an PseudoSQL statement and provides methods for accessing them. For parsing the PseudoSQL, the original parser written by Helmut Lichtenberg <heli@tzv.fal.de> was used.

Public and internal methods are:

PUBLIC METHODS

actionname - returns the sql action

tablename - returns the table name used in the statement - only one table is allowed per statement!

columns

- returns list of column names

values - returns list of column values

extfields - returns list of external fields that are targeted for errors

value - returns the value of the supplied column

column_extfields - returns list of external fields for a certain column

whereclause - returns the where part of the statement

status - returns the object status - inherited from apiis

errors - returns list of error object - inherited from apiis

_standard_keys (internal) - encapsulates the names of the automatically created methods

_accessible (internal) - checks if the method is read-only or read-write

new - creates the parsed sql object

`_init (internal)` - calls `_ParsePseudoSQL` for parsing of the SQL and fills the structure

`pull_quotes`

`pull_quotes -- tom christiansen, tchrist@convex.com`

AUTHORS Zhivko Duchevev <duchev@tzv.fal.de> Helmut Lichtenberg <heli@tzv.fal.de>

6.6.8.4 Statement

SYNOPSIS

```
$statement = Apiis::DataBase::SQL::Statement->new(
    sql      => $sqltext
);
```

This is the module for creating an object for parsed normal SQL. This module is intended for parsing simple SQL statements: INSERT, UPDATE, DELETE, SELECT (without aggregate functions).

DESCRIPTION Creates the internal structure for important elements of SQL statement and provides methods for accessing them

Public and internal methods are:

PUBLIC METHODS

`actionname` - returns the sql action

`tablename` - returns the table name used in the statement - only one table is allowed per statement!

`columns`

- returns list of column names

`values` - returns list of column values

`value` - returns the value of the supplied column

`whereclause` - returns the where part of the statement

`status` - returns the object status - inherited from apiis

`errors` - returns list of error object - inherited from apiis

`_standard_keys (internal)` encapsulates the names of the automatically created methods

_accessible (internal) checks if the method is read-only or read-write

new - creates the parsed sql object

_init (internal) - parses the SQL and fills the structure

AUTHORS Zhivko Ducheve <duchev@tzv.fal.de>

6.6.8.5 Apiis::DataBase::MakeSQL Module to create SQL-statements from the model file

DESCRIPTION **MakeSQL** as the main method and some auxiliary routines read the Apiis model file and create a file with SQL data definition commands to create tables, views, indices, sequences, etc.

The resulting file will either be written to STDOUT or placed in the var-subdirectory of the given project. Its name is created from the project name with database driver and .sql extension appended:

```
<project>_<db_driver>.sql
breedprg_Pg.sql
```

METHODS Besides the main method **MakeSQL** there are the auxiliary methods **Cascaded_FK**, **resolve_concatenations**, and **HasFKRule**.

Read 'perldoc \$APIIS_HOME/bin/mksql' for the most prominent implementation and for detailed usage information.

6.6.9 The Synchronization

6.6.9.1 SYNOPSIS

```
$node = Apiis::DataBase::Sync::Node->new(
                                                    nodename=>'Mariensee',
                                                    class_column=>'owner'
                                                    );
```

This is the module for creating an object that handles the node information
 . =head1 DESCRIPTION

Provides methods to create new server or client of the synchronization process, reading DataElementDescriptions, reading data state, comparing states etc.

Public and internal methods are:

6.6.9.2 PUBLIC METHODS

new - creates new node object

_init (internal) -creates dynamic methods and fill the information about all registered nodes

create_client - creates new Net::EasyTCP client Usage: \$node->create_client(\$server_host,\$port)

create_server - creates new Net::EasyTCP server Usage: `$node->create_server($port)`

read_DED - reads DataElementDescriptions (table,class,column names) for node `$node_name` and role source or target from the database. Usage: `$node->read_DED($node_name,'source')` or `$node->read_DED($node_name,'target')`

ip2name - converts IP address into name accordingly to the information in 'nodes' table Usage: `$node->ip2name($nodeip)`

name2ip - converts name into IP address accordingly to the information in 'nodes' table Usage: `$node->name2ip($nodename)`

_loadnodes (internal) - reads from the database information about all registered nodes

DED_state - sets or reads the state of ONE DataElement Usage: `$node->DED_state(\%data_element_state)` or `$node->DED_state`

compare_states - compares the internal state in `DED_state` with the one passed as a parameter and replaces the version information in the passed hash with merge action info. In case of success returns also reference to the modified hash Usage: `$node->compare_states(\%data_element_state)`

load_merge_element - Create SQL statement from the DED and merge element and executes it Usage: `$node->load_merge_element($DED,$merge_element)`

6.6.9.3 AUTHORS

Zhivko Duchevev <duchev@tzv.fal.de>

6.7 The Forms

6.7.1 Apiis::Form::Init – base package for Form objects of all types

SYNOPSIS

This base package provides the main functionality and methods needed for all Form object types (Tk, Html, etc.).

DESCRIPTION

The generic public methods of this base class are described below. Most of them can be inherited by subsequent Form objects.

METHODS

exists_fieldtype (public) Subroutines to check, if the passed type exists in the list of hardcoded fieldtypes.

exists_ds_type (internal) Returns true if the passed parameter is in the list of hardcoded DataSource Types (like sql, record, function, none).

is_a_listfield (internal) Returns true if the passed parameter is in the list of hardcoded fields, that have a list character (like ScrollingList, BrowseEntry, etc.).

is_misc_blockelement (internal) Subroutines to check, if the passed element exists in the list of hardcoded Block elements, which are not Field, DataSource, etc..

new (public) **new** creates a new form object. Input parameter is an anonymous hash with the key 'xmlfile' and the path to this file as its value.

The **new** method of Init.pm is not invoked directly but via inheritance through the widget specific modules.

Example:

```
my $form_obj = Apiis::Form::Tk->new(
    xmlfile => '/path/to/xmlfile.frm'
);
```

_init (internal) **_init()** is only invoked if you want to create an object of type Apiis::Form directly. As this is only a base class an error is yielded.

xmlfile (public) **xmlfile()** returns the full path of the xml file for this form.

add_formlib_path (public) **add_formlib_path** will prepend some Form/Project specific directories to the library search for Form modules in up to down order. So the most specific will be searched first and wins over the more generic ones. Form modules for example for Form_0 and GUI-type Tk are searched in:

```
$APIIS_HOME/lib/Apiis/Form/Tk/Form_0/
$APIIS_HOME/lib/Apiis/Form/Tk/
$APIIS_HOME/lib/Apiis/Form/Form_0/
$APIIS_HOME/lib/Apiis/Form/Event/
```

As Form-name and GUI-type are determined at runtime, the addition of this search paths also have to happen at runtime.

add_formlib_path expects as input parameters:

1. the GUI-type (e.g. Tk, HTML, Qt)
2. the form name as in `$self->formname`

Usage:

```
$self->add_formlib_path( 'Tk', $self->formname );
```

form_status_msg (public) **form_status_msg** is a write-only method to put form-global status messages somewhere. They can be displayed e.g. in a status line (textfield) in the form.

Example:

```
$self->form_status_msg('Retrieved 20 records');
```

GetValue (public) **GetValue** is the main method to access the configuration data and the data references of this form object. All the real elements (not the pseudo auxiliary elements) of the xml file have a unique name and are accessible through this name. All elements without the identifying name are flattened into their parent element. Their attributes become attributes of the parent.

Syntax:

```
my $attr_value = $form_obj->GetValue($elementname, $attribut);
```

Example:

```
my $label = $form_obj->GetValue('Field_1', 'Label');
```

GetValue returns also the values for dynamically created items like `'_field_list'` for each block or `'_column_list'` for each datasource.

Example:

```
foreach my $field ( @{ $form_obj->GetValue( 'Block_0', '_field_list' ) } ){
    # do something
}
```

The list values return an array reference so they must be dereferenced for usage in loops.

SetValue (public) **SetValue** is the counterpart to **GetValue**.

Syntax:

```
my $old_value = $form_obj->SetValue($elementname, $attribut, $new_value);
```

SetValue returns the old value.

IncValue | DecValue (public) Increment and decrement values in the flattened structure.

Syntax:

```
$form_obj->IncValue($elementname, $attribut);
$form_obj->DecValue($elementname, $attribut);
```

GetEvent (public) **GetEvent** should make access to event definitions at certain levels easy.

Syntax:

```
my $events_ref = $form_obj->GetEvent($elementname, 'OnClick');
```

GetEvent returns an arrayreference to all `'OnClick'`-Eventnames of `$elementname` or `undef` if none exists.

RunEvent (public) input: hash reference with required keys: elementname, eventtype

output: array reference with the names of the processed events

gui_type (public) **gui_type** returns (and sets) the type of the running Graphical User Interface, e.g. Tk, HTML.

top (public) **top** returns (and sets) a reference to toplevel window (Tk, query => HTML).

formname (public) **formname** returns the name of the Form.

For the consistency of method names, there is also a method **formnames**, which returns an arrayref of the list of formnames.

Both methods might be seldom used.

generalname (public) **generalname** returns the name of the General section. There could only be one General section.

For the consistency of method names, there is also a method **generalnames**, which returns an arrayref of the list of generalnames.

blocknames | datasourcenames | fieldnames | columnnames (public)

These methods all return an arrayref of the list of names.

misc_blockelements (public) **misc_blockelements** return an array (or reference) to a list of non-Field elements of a block (Line, Image, Frage, etc.).

master_detail (public) Form level flag, if this form is a more complex one with master/detail relationships.

Usage:

```
$self->master_detail(1); # set flag
&easy_going if not $self->master_detail;
```

query_block_order (public) **query_block_order** return the blocks in the right order for Master/Detail handling. The query order depends on the master/detail relationships between blocks. This order is determined in Event/Query.pm, when the first query is started. After this it is usually used readonly.

encode_list_ref | decode_list_ref **encode_list_ref** and **decode_list_ref** exchange the primary/foreignkey value with the more readable one and vice versa. These methods are used, if a field-specific DataSource exists.

Both take as input parameters:

1. Fieldname
2. The data to en/de-code

They return the en/de-coded value.

fieldtype (internal) **fieldtype** returns the widget-set specific fieldtype for a passed metatype.

Example:

```
my $ft = $self->fieldtype('textfield'); # returns 'TextField' for Tk
```

All metatypes are in lower case.

To make error messages more helpful, you should add a second parameter, the fieldname:

```
my $ft = $self->fieldtype( $type, $fieldname );
```

If no error occurs, the fieldname is simply ignored, otherwise it will give a valuable hint, where to search in the XML file.

insert_block | insert_form | update_block | query_block | clear_block | clear_form (public) These methods handle form events on block level, initiated by the user. These are the public interfaces, the real methods are rolled out into the `Apiis::Form::Event` namespace. See details there.

The blockname is provided during the invocation of the `do_<commands>` in the widget-specific button handling, e.g. in `Tk/Button.pm`.

insert_blocks | insert_form | clear_form (public) These are the same methods except that they act on multiple/all blocks. The blocknames for **insert_blocks** are defined in the xml file.

next | prev | first | last (public) These methods are for navigation through the records while querying data.

return_value (public) **return_value** stores and gives back a SCALAR value, which could be either a simple scalar, a reference to a hash, to an array or even to another object. It's up to the configuration, what type of return value is expected.

CreateCSSProperties

This function has an object name as argument. It fetches the object about `getobject($name)` and make a loop over the `css-children-objects`. After test whether the children-objects (color, format, position, miscellaneous, text) are valid each methodes of the objects were ask if it has an entry. About hash `%css` the correct css-name will fetched. After loop all properties were collect and return as `STYLE`-element

6.7.2 The Forms old

popup

popup opens a popup window with some informational or error message and waits for a notification by the user.

popup is a slightly changed derivat from Hartmut's `yaform` version.

Usage:

```

popup(
    {   toplevel => $top,
        title    => $title,
        bitmap   => $bitmap,
        text     => $text,
        buttons  => $buttons_ref,
    }
);

```

6.8 The Misc

6.8.1 usage

`get_value('position cfix position2', \@lineref, format)`

6.8.2 description

return value from datafile at specific position @lineref is the line from datafile
 return: value if position known as code (from collect_codes1.pl) return db_code
 after check file codes.chg for possible changing this code same for unit and
 table_id MUST be used if fields can be empty: as a consequence the variable
 is not an empty string but a NULL value!!!

6.8.3 configuration

position

describe the position on input line. could be more than once and also fixed
 parts are possible (see the c-option).

format

define either an date format, a simple test if the return value is an number or
 lower resp. upper case any character.

[dmyj] plus any character which split the date (./...) date format
 use the function `getdate()`. this allow an specific format like 'dd.mm.yyyy' or
 'ddmmjj' or use the number of elements in the data. for further details see
 function `getdate()`.

n test is a number else ignore the value. also change from ',' to '.' if exist.

cl lower case characters.

cu upper case characters.

6.8.4 used in

loading historic data (`load_data.pl`)

6.8.5 usage

getdate(date) or getdate(date, format)

6.8.6 return

formatted date, status, err_msg

6.8.7 description

getdate should be simplify the handling of dates in incomming datastreams.
only the following dates are possible:

1. 799 -> 3-Juli-1999
2. 799 -> 12-Juli-1999
3. -> 12-Dezember-1999

12.07.99|12-07-99|12:07:99 -> 12-Juli-1999

or using a format as second parameter like 'dd.mm.jjjj' or 'yyy.tt.mm' or 'ttmmjj' and so on
getdate('19984/02','yyyymm/tt') => 2-April-1998

6.8.8 xfig_lib.pm

Library for model2xfig

DESCRIPTION

xfig_lib initialized basic variables, sets default values and defines subroutines for creating the FIG-header, for compounding objects, drawing lines, arrowlines, boxes and text.

SUBROUTINES

getFileHeader getFileHeader creates the FIG-file header.

usage: \$header = getFileHeader([comment])

comment: comment to print into the header

returnvalue: string with header

texttype creates a *text* object.

usage: \$textobj = texttype(\$text, \$type, \$xpos, \$ypos, \$depthT)

text: string

type: t = title; c = columnname ; h = head line

xpos: x-position

ypos: y-position

depthT: depth (layer)

returnvalue: string with text object

boxtype creates a *box* object.

usage: \$boxobj = boxtype(\$x1, \$y1, \$x2, \$y2)

x1,y1: first corner point

x2,y2: final corner point

returnvalue: string with box object

linetype creates a *line* object.

usage: \$lineobj = linetype(\$x1, \$y1, \$x2, \$y2)

x1,y1: first point

x2,y2: final point

returnvalue: string with line object

arrowlinetype creates a *arrowline* object with a **backward** arrow.

usage: \$arrowlineobj = arrowlinetype(\$x1, \$y1, \$x2, \$y2)

x1,y1: first point

x2,y2: final point

returnvalue: string with arrowline object

compoundtype creates a *compound* object.

usage: \$compoundobj = compoundtype(\$objects, \$upperright_corner_x,
\$upperright_corner_y, \$lowerleft_corner_x, \$lowerleft_corner_y)

objects: string with xfig objects. Created with
texttype, boxtype, linetype and
arrowlinetype

upperright_corner_x, upperright_corner_y,
lowerleft_corner_x, lowerleft_corner_y
defines a rectangle so that all objects are
inside of this region.

returnvalue: string with compound object

SEE ALSO

model2xfig, xfig (www.xfig.org)

AUTHOR

Hartmut Börner (haboe@tzv.fal.de)

6.9 The Binaries

6.9.1 check_integrity

SYNOPSIS

check_integrity -p <project> [Options]

OPTIONS

```

-p | --project <project>  defines the project to check (r)

-u | --user <user>        provide username <user> to connect to project (o)
-P | --password <passwd>  provide password <passwd> to connect to project (o)

-t | --table <table>      check only this table. You can provide several -t
                           options or a list of comma separated tables (o)
-s | --stop <number>      stop checking after <number> records (o)

-e | --errfile             write detailed errors into files of the form
                           <project>_<table>.err (o)
-f | --filename <file>    writes summary of check results into <file>
                           instead of the default file check_integrity.err. (o)
-D | --dirtyflag           use column 'dirty' to flag checked records (o)
-S | --skipdirty           skip records which are already flagged as dirty (o)

-h | -? | --help          short help (o)
-m | --man                detailed man page (o)
-v | --version            current version of check_integrity (o)

(r) - required, (o) - optional

```

DESCRIPTION

check_integrity checks your database for data integrity according to the defined rule in the project's model file.

The option **-p <project>** is the only required one.

The **-t <table>** options allows specifying either a single table or several tables in a comma separated list. The list must not contain any blanks. You can also give several **-t <table>** options on the command line.

EXAMPLES

```

check_integrity -p breedprg
check_integrity -p breedprg -u demo -P 'my secret' -t animal -s 100
check_integrity -p breedprg -t animal -t transfer -t codes -D -S
check_integrity -p breedprg -t animal,transfer,codes -DSe -f check_report.txt

```

BUGS

check_integrity is slow. The main area to speed it up will be an optimization of the Record objects.

VERSION

\$Revision: 1.54 \$

AUTHOR

```

Helmut Lichtenberg <heli@tzv.fal.de>
Ralf Fischer <ralf@tzv.fal.de>

```


6.9.2 cvs2cl.pl

Convert cvs log messages to changelogs

SYNOPSIS

cvs2cl [*options*] [*FILE1* [*FILE2* ...]]

DESCRIPTION

cvs2cl produces a GNU-style ChangeLog for CVS-controlled sources by running "cvs log" and parsing the output. Duplicate log messages get unified in the Right Way.

The default output of cvs2cl is designed to be compact, formally unambiguous, but still easy for humans to read. It should be largely self-explanatory; the one abbreviation that might not be obvious is "utags". That stands for "universal tags" – a universal tag is one held by all the files in a given change entry.

If you need output that's easy for a program to parse, use the **-xml** option. Note that with XML output, just about all available information is included with each change entry, whether you asked for it or not, on the theory that your parser can ignore anything it's not looking for.

If filenames are given as arguments cvs2cl only shows log information for the named files.

OPTIONS

-h, -help, --help, -?

Show a short help and exit.

--version

Show version and exit.

-r, --revisions

Show revision numbers in output.

-b, --branches

Show branch names in revisions when possible.

-t, --tags

Show tags (symbolic names) in output.

-T, --tagdates

Show tags in output on their first occurrence.

--show-dead

Show dead files.

--stdin

Read from stdin, don't run cvs log.

-stdout

Output to stdout not to ChangeLog.

-d, -distributed

Put ChangeLogs in subdirs.

-f *FILE*, -file *FILE*

Write to *FILE* instead of ChangeLog.

-fsf

Use this if log data is in FSF ChangeLog style.

-FSF

Attempt strict FSF-standard compatible output.

-W *SECS*, -window *SECS*

Window of time within which log entries unify.

-U *UFILE*, -usermap *UFILE*

Expand usernames to email addresses from *UFILE*.

-passwd *PASSWORDFILE*

Use system passwd file for user name expansion. If no mail domain is provided (via **-domain**), it tries to read one from **/etc/mailname**, output of **hostname -d**, **dnsdomainname**, or **domain-name**. cvs2cl exits with an error if none of those options is successful. Use a domain of " to prevent the addition of a mail domain.

-domain *DOMAIN*

Domain to build email addresses from.

-gecos

Get user information from GECOS data.

-R *REGEXP*, -regexp *REGEXP*

Include only entries that match *REGEXP*. This option may be used multiple times.

-I *REGEXP*, -ignore *REGEXP*

Ignore files whose names match *REGEXP*. This option may be used multiple times. The regexp is a perl regular expression. It is matched as is; you may want to prefix with a **^** or suffix with a **\$** to anchor the match.

-C, -case-insensitive

Any regexp matching is done case-insensitively.

-F *BRANCH*, -follow *BRANCH*

Show only revisions on or ancestral to *BRANCH*.

-follow-only *BRANCH*

Like `-follow`, but sub-branches are not followed.

-no-ancestors

When using `-F`, only track changes since the *BRANCH* started.

-no-hide-branch-additions

By default, entries generated by cvs for a file added on a branch (a dead 1.1 entry) are not shown. This flag reverses that action.

-S, -separate-header

Blank line between each header and log message.

-summary

Add CVS change summary information.

-no-wrap

Don't auto-wrap log message (recommend `-S` also).

-no-indent

Don't indent log message

-gmt, -utc

Show times in GMT/UTC instead of local time.

-accum

Add to an existing ChangeLog (incompatible with `-xml`).

-w, -day-of-week

Show day of week.

-no-times

Don't show times in output.

-chrono

Output log in chronological order (default is reverse chronological order).

-header *FILE*

Get ChangeLog header from *FILE* ("`-`" means stdin).

-xml

Output XML instead of ChangeLog format.

-xml-encoding *ENCODING*.

Insert encoding clause in XML header.

-noxmlns

Don't include `xmlns=` attribute in root element.

-hide-filenames

Don't show filenames (ignored for XML output).

-no-common-dir

Don't shorten directory names from filenames.

-rcs *CVSROOT*

Handle filenames from raw RCS, for instance those produced by "cvs rlog" output, stripping the prefix *CVSROOT*.

-P, -prune

Don't show empty log messages.

-lines-modified

Output the number of lines added and the number of lines removed for each checkin (if applicable). At the moment, this only affects the XML output mode.

-ignore-tag *TAG*

Ignore individual changes that are associated with a given tag. May be repeated, if so, changes that are associated with any of the given tags are ignored.

-show-tag *TAG*

Log only individual changes that are associated with a given tag. May be repeated, if so, changes that are associated with any of the given tags are logged.

-delta *FROM_TAG TO_TAG*

Attempt a delta between two tags (since *FROM_TAG* up to and including *TO_TAG*). The algorithm is a simple date-based one (this is a hard problem) so results are imperfect.

-g *OPTS*, -global-opts *OPTS*

Pass *OPTS* to cvs like in "cvs *OPTS* log ...".

-l *OPTS*, -log-opts *OPTS*

Pass *OPTS* to cvs log like in "cvs ... log *OPTS*".

Notes about the options and arguments:

- The **-I** and **-F** options may appear multiple times.
- To follow trunk revisions, use **"-F trunk"** (**"-F TRUNK"** also works). This is okay because no would ever, ever be crazy enough to name a branch "trunk", right? Right.
- For the **-U** option, the *UFILE* should be formatted like CVSROOT/users. That is, each line of *UFILE* looks like this:

```
jrandom:jrandom@red-bean.com
```

or maybe even like this

```
jrandom: 'Jesse Q. Random <jrandom@red-bean.com>'
```

Don't forget to quote the portion after the colon if necessary.

- Many people want to filter by date. To do so, invoke `cvs2cl.pl` like this:

```
cvs2cl.pl -l "-d'DATESPEC'"
```

where `DATESPEC` is any date specification valid for `"cvs log -d"`. (Note that CVS 1.10.7 and below requires there be no space between `-d` and its argument).

- Dates/times are interpreted in the local time zone.
- Remember to quote the argument to `'-l'` so that your shell doesn't interpret spaces as argument separators.
- See the 'Common Options' section of the `cvs` manual (`'info cvs'` on UNIX-like systems) for more information.
- Note that the rules for quoting under windows shells are different.
- To run in an automated environment such as CGI or PHP, `suidperl` may be needed in order to execute as the correct user to enable `/cvsroot` read lock files to be written for the `'cvs log'` command. This is likely just a case of changing the `/usr/bin/perl` command to `/usr/bin/suidperl`, and explicitly declaring the `PATH` variable.

EXAMPLES

Some examples (working on UNIX shells):

```
# logs after 6th March, 2003 (inclusive)
cvs2cl.pl -l "-d'>2003-03-06'"
# logs after 4:34PM 6th March, 2003 (inclusive)
cvs2cl.pl -l "-d'>2003-03-06 16:34'"
# logs between 4:46PM 6th March, 2003 (exclusive) and
# 4:34PM 6th March, 2003 (inclusive)
cvs2cl.pl -l "-d'2003-03-06 16:46>2003-03-06 16:34'"
```

Some examples (on non-UNIX shells):

```
# Reported to work on windows xp/2000
cvs2cl.pl -l "-d"">2003-10-18;today<"""
```

AUTHORS**Karl Fogel****Melissa O'Neill****Martyn J. Pearce**

Contributions from

Mike Ayers**Tim Bradshaw****Richard Broberg****Nathan Bryant****Oswald Buddenhagen****Neil Conway****Arthur de Jong****Mark W. Eichin****Dave Elcock****Reid Ellis****Simon Josefsson****Robin Hugh Johnson****Terry Kane****Pete Kempf****Akos Kiss****Claus Klein****Eddie Kohler****Richard Laager****Kevin Lilly****Karl-Heinz Marbaise****Mitsuaki Masuhara****Henrik Nordstrom****Joe Orton****Peter Palfrader****Thomas Parmelan**

Jordan Russell

Jacek Sliwerski

Johannes Stezenbach

Joseph Walton

Ernie Zapata

BUGS

Please report bugs to `bug-cvs2cl@red-bean.com`.

PREREQUISITES

This script requires `Text::Wrap`, `Time::Local`, and `File::Basename`. It also seems to require `Perl 5.004_04` or higher.

OPERATING SYSTEM COMPATIBILITY

Should work on any OS.

SCRIPT CATEGORIES

Version_Control/CVS

COPYRIGHT

(C) 2001,2002,2003,2004 Martyn J. Pearce <fluffy@cpan.org>, under the GNU GPL.

(C) 1999 Karl Fogel <kfogel@red-bean.com>, under the GNU GPL.

`cvs2cl.pl` 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, or (at your option) any later version.

`cvs2cl.pl` 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 may have received a copy of the GNU General Public License along with `cvs2cl.pl`; see the file `COPYING`. If not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

SEE ALSO

`cvs(1)`

6.9.3 file2inspool.pl

The program is looking for files in directory "\$APIIS_LOCAL/load/inspool", reads the data and loads them into the INSPool table in your database. Files in the inspool directory have to be text files containing one record per line.

Each filename in your inspool directory has to start with DSxx where xx is the number of the data stream (DS01, DS02, ...)

After running the program, each row of the INSPool table contains the data stream identification (DS01, DS02...) in column ds, a unique number generated by sequence 'seq_inspool__in_id' (column in_id), the status (NEW), the time of processing (timestamp) and the whole, unchanged record in ASCII format (column record).

The table LOAD_STAT is populated with a new record about the insertion of the new datastream into INSPool. The filename of the loaded data is stored in column JOB, the start and end of this job are recorded in the appropriate columns, also the total number of inserted records (nrec_tot).

After successful loading the datafile is moved to "\$APIIS_LOCAL/load/inspool/done".

If the datastream contains BLOBS (binary large objects) - pictures, movies etc. then the new special table "blobs" is used. In the datastream on the third row - the word "blobs" should occur followed by the number of elements in a row and position number of the file names in the data rows. For example:

```
DS01
ini
blobs 5 1 3
cat|/home/zgr/duchev/pictures/IN00006A.JPG|123.56|IN00009A.JPG|jpg
dog|IN00004A.JPG|87.10|/home/zgr/duchev/pictures/IN00005A.JPG|jpg
```

All files from one datastream should be stored in separate subfolder of the "inspool" folder. The "file2inspool.pl" should be invoked in the following manner:

```
file2inspool.pl -f <folder name> -m <model_file>
```

The script reads DS file from this folder, loads files into BLOBS table, stores the returned record identifiers instead of the file names in the DS file and load these records as usual in the INSPool table.

6.9.4 Form

Create a window with the form defined in formfile

SYNOPSIS

```
Form [-s][-d][-D level][-p][-P pos] [formfile] [data ... ]
```

DESCRIPTION

The formfile describe the appearance and functionality of a form. **Form** initializes necessary things and passes the formfile and parameter to the module 'yaform.pm'. For using more than one form at a time one can also use the wrapper **apiish**.

To pass data directly into fields of the form one can give the data to the command line. The first given parameter after formfile will be written into the first field of the form, the second into the second field and so on. One can check

the order of the form field with the FormDesigner at the Edit menu. With option -p and the data parameter it is possible to pass values from one form to an other.

OPTIONS

-s

print last SQL Statement

-d

print debug messages from yaform

-D level

DBIx::Recordset debuglevel 1..4

-P

print field values from the returned hash

-P pos

place in one of seven screen positions (default: 0) pos: .———. |1 2 3| | 0 |
|4 5 6| '———'

SEE ALSO

yaform.pm, form_ulib.pm, FormDesigner, apiish

AUTHOR

Hartmut Börner (haboe@tzv.fal.de)

6.9.5 FormDesigner

GUI for creating/editing formfiles

SYNOPSIS

FormDesigner

DESCRIPTION

FormDesigner is a graphical user interface for creating and editing form files (formfile) for APIIS (Adaptable Platform Independent Information System).

The bulk of the documentation for *FormDesigner* is in an HTML-based implementer guide. See the Help menu in *FormDesigner* or point your browser at *\$APIIS_HOME/doc/implementer/FormDesigner/de/FormDesigner.html*. At this time, there is only a german version available.

For the help system the preferred browser can be permanently set in the apiis configuration file *\$APIIS_HOME/apiisrc*

SUBROUTINES

newForm newForm initialize or reset some variables for a new Form. If a form exists ask about destroying, then create a new one. Creates the GENERAL-configuration window.

Subroutine is called form the main window menu: File->New and Edit->GENERAL

usage: newForm(\$ToplevelWindow [, \$switch])

ToplevelWindow: created with i.e. \$ToplevelWindow = MainWindow->new();

switch: if set, create only the GENERAL-configuration window.

loadForm loadForm loads a existing formfile for editing.

usage: loadForm(\$ToplevelWindow)

saveForm saveForm stores the form definitions to a formfile.

usage: saveForm(\$ToplevelWindow [, \$Mode])

ToplevelWindow: toplevel widget

Mode: if set, ask about filename.
if not set, store either to untitled.form or
to a previously given filename.

loadmodel Load model from given modelfile.

usage: loadmodel(\$ToplevelWindow [, \$nFwin])

ToplevelWindow: toplevel widget

nFwin: GENERAL-configuration toplevel widget
if given, destroy widget

selectFile selectFile opens a file selector box to choose a filename for loading or saving. Two different file selector boxes are available. Tk::FBox (default) and Tk::FileSelect. In *\$APIIS_HOME/apiisrc* one can overwrite the default by setting the \$fileselector variable to 'FileSelect'.

usage: \$filename = selectFile(\$operation, \$ToplevelWindow)

operation: one of four operations

model: to open a modelfile
loadform: to open a formfile
saveform: to save a formfile
gif: to open a GIF-image

ToplevelWindow: toplevel widget

selectFont selectFont is a font browser and chooser for X Window fonts. Therefore only available for unix operating systems.

usage: selectFont(\$ToplevelWindow, \$switch, \$font_ref)

ToplevelWindow: toplevel widget
 switch: one of six settings. It is only used to set the title of the font browser window.
 0: "Select: Title Font"
 1: "Select: Normal Font"
 2: "Select: Label Font"
 3: "Select: Button Font"
 4: "Select: Font for digital time"
 5: "Select: Font for digital date"
 font_ref: a reference to the name of the X font string

warnwin warnwin creates a modal dialogbox window and waits for response.

usage: \$answer = warnwin(\$ToplevelWindow, \$title, \$bitmap, \$text, [\@buttons])

ToplevelWindow: toplevel widget
 title: title of the dialogbox window
 bitmap: specifies a bitmap to display in the top portion of the dialog, to the left of the text. If this is an empty string then no bitmap is displayed in the dialog. There are a set of Tk build-in bitmaps. The most used are 'error', 'info', 'question', 'questhead' and 'warning'.
 bitmap can also be a Tk::Pixmap object.
 text: the text to display on the dialogbox
 buttons: reference to an array with the buttons text.
 i.e. ['Ok', 'Cancel'] (two button)
 ['Red', 'Green', 'Blue'] (three button)

Example: # what would you do?

```
$answer = warnwin($top,           # toplevel window
                  'Question',      # window title
                  'questhead',     # predefined bitmap
                  'What would you do?', # text
                  ['work', 'play', 'sleep']); # three buttons
if($answer eq 'work') { # do something }
if($answer eq 'play') { system("battlechess" }
if($answer eq 'sleep') { sleep(28800) }
```

clearGENERAL clearGENERAL clears all entrys in the GENERAL-configuration window

usage: clearGENERAL(\$ToplevelWindow, \$GeneralWindow)

ToplevelWindow: toplevel widget
 GeneralWindow: widget of the GENERAL-configuration window

fillColList fillColList fills for a given table in the column listbox the existing column names. After choosing a table this subroutine is executed.

usage: fillColList(\$table-Listbox-widget);

table-Listbox-widget: widget of the table listbox.

It's used to retrieve the selected table
from the table listbox.

proceed proceed sets formfile section name and field type.

usage: proceed(\$ToplevelWindow)

ToplevelWindow: toplevel widget

config_EDLBNCATIOPMRU creates for each field types a different configuration window. Only necessary parameters will be shown.

usage: config_EDLBNCATIOPMRU(\$ToplevelWindow)

ToplevelWindow: toplevel widget

config_unknown usage: config_unknown(\$ToplevelWindow)

ToplevelWindow: toplevel widget

placeField register a section and place it on the form

usage: placeField(\$ToplevelWindow)

ToplevelWindow: toplevel widget

editsection open a registered section for editing

usage: editsection(\$section)

section: section name to be edited

updateField if a Field is placed then this is the right subroutine to create or update the Form\().

usage: updateField()

deleteField delete a registered section

usage: deleteField(\$section)

section: a registered section to delete

textEdit textEdit is a small text editor for editing parameters that could be more than one line long.

usage: textEdit(\$section, \$parameter, \$ToplevelWindow)

section: a registered section

parameter: parameter of section

ToplevelWindow: toplevel widget

options creates a window for selecting a browser
usage: options(\$ToplevelWindow)

ToplevelWindow: toplevel widget

option_debug set debug option. Used to print some debug messages from yaform.pm
usage: option_debug()

option_sql set option 'last SQL statement'. Used to print some SQL messages from yaform.pm
usage: option_sql()

option_apiis option_apiis creates a window to set or edit \$APIIS_HOME and \$APIIS_LOCAL.
usage: option_apiis(\$ToplevelWindow)

ToplevelWindow: toplevel widget

option_grid option_grid creates a window to set x/y grid values.
usage: option_grid(\$ToplevelWindow)

ToplevelWindow: toplevel widget

opt_fsel creates a window to select a fileselector (FBox/FileSelector).
usage: opt_fsel(\$ToplevelWindow)

ToplevelWindow: toplevel widget

renSection renSection creates a window to rename a formfile section
usage: renSection(\$section, \$ToplevelWindow)

section: section name to delete

ToplevelWindow: toplevel widget

checkSec checkSec is called from subroutine renSection. Makes the extensive stuff to check of existing sections, changes the form hash and the registration, adjusts the edit menu, renames some window titles and button text.
usage: checkSec(\$cursec, \$newsec, \$toplevel)

cursec: current section name to be renamed

newsec: the new section name

toplevel: toplevel window ('Rename Section' window)
will be destroyed after successfull renaming.

Post Slaven Rezac has made a nice patch that solves the problem of slow Dialog boxes perfectly. Here is his solution.
thanx Slaven!

SEE ALSO

`$APIIS_HOME/lib/yaform.pm`, `$APIIS_HOME/lib/form_ulib.pm`

AUTHOR

Hartmut Börner (haboe@tzv.fal.de)

6.9.6 mkform

Create a formfile for each table in the database

SYNOPSIS

```
mkform [-y deltaY][-c column1[,column2 ...]] <modelfile> mkform [-h]
```

DESCRIPTION

The program `mkform` creates a GUI parameter file for each table in the database assuming inserts. This procedure can be used to quickly generate an application for simple problems. The resulting form files can be edited manually.

Reads the modelfile from the current working directory or from directory `$APIIS_LOCAL/model/`

The formfiles will be written either to `dirname(<modelfile>)/default_forms`, `$APIIS_HOME/model/default_forms` or in an subdirectory `default_forms` of the current directory. The name of the created formfile is `<table>.frm`

OPTIONS**-y vertical field spacing.**

Formfile parameter YLOCATION (default 50)

-c omit column**-h short help****FILES**

```
$APIIS_LOCAL/<model>.model
$APIIS_HOME/apiisrc
$HOME/.apiisrc
```

AUTHOR

Hartmut Börner (haboe@tzv.fal.de)

mkLOfForm -

6.9.7 create a formatted form file from each given loadobject by parsing the loadobject file about variable @LO_ keys

SYNOPSIS

```
mkLOfForm [-y] [-f] <load_object_files>  
mkLOfForm -h | -v
```

-y vertical field spacing.

Formfile parameter YLOCATION (default 48)

-f fieldlength

fieldlength of each field (default 25)

-v version

-h help

EXAMPLE

```
mkLOfForm LO_*
```

SEE ALSO

mkform, mkLOform

AUTHOR

Hartmut Boerner (haboe@tzv.fal.de)

6.9.8 mkLOform

Create a form file from each given loadobject

SYNOPSIS

```
mkLOform [-y] [-f] <load_object_files> <modelfile>
```

-y vertical field spacing.

Formfile parameter YLOCATION (default 48)

-f fieldlength

fieldlength of each field (default 45)

-v version

-h help

EXAMPLE

```
mkLOform LO_* '$APIIS_LOCAL/model/apiis.model'
```

SEE ALSO

mkform

AUTHOR

Hartmut Börner (haboe@tzv.fal.de)

6.9.9 mksql – create SQL commands from the model file**SYNOPSIS**

```
mksql [-h|v|t|d|n|s] [-f] <model file>
```

Create SQL commands from the model file to create the database structure

OPTIONS

-h	Help
-m	show manpage
-v	Version
-f <modelfile>	Name of model file (required)
-t <table>	only for table <table>
-d	delete: DROP-statements are not commented out!
-n	create no views
-s	write to STDOUT

DESCRIPTION

mksql reads the model file and writes a SQL-file (unless -s) to create all necessary tables, view, indexes, sequences, etc. As this is very database specific you have to run this file by hand. Maybe you want to inspect it before. :^)

SEE ALSO

Apiis::DataBase::SQL::MakeSQL;

COPYRIGHT

This program is free software; you can redistribute it and/or modify it under the same terms as Perl itself.

See <<http://www.perl.com/perl/misc/Artistic.html>>

AUTHOR

Helmut Lichtenberg <heli@tzv.fal.de>

6.9.10 model2xfig

Create a database relationship diagram

SYNOPSIS

```
model2xfig [options] [modelfile]
```


DESCRIPTION

model2xfig creates a relationship diagram from a modelfile in xfig file format (FIG 3.2). The name of the xfig file is *modelfile.fig*.

After loading the file into **xfig** one can move tables with 'Smart Links' set to 'MOVE'. In this mode the end of lines (foreign keys) persists.

OPTIONS**-h**

help

-v

version

-s

stack tables - put tables in layers one on top of the other. The order of the stacked tables correspondence to the order of the defined tables in the model file. The last defined table lies on top of the others.

-R <yes | no>

relations yes: lines with arrows to indicate foreign keys (default) no: no lines for foreign keys

-b <fill | nofill | value>

box fill fill: opaque table boxes, color white nofill: (default) transparent table boxes, visible crossing lines within the boxes. value: -1 = not filled, same as 'nofill' 0 = black 1-19 shades of grey, from darker to lighter 20 = white, same as 'fill' 41-56 patterns(?)

SEE ALSO

xfig (www.xfig.org)

AUTHOR

Hartmut Börner (haboe@tzv.fal.de)

6.9.11 show__rules**usage**

show__rules -h -v -o <output file> -p <projectname>

description

used to compare some values from the parents with the same trait from the animal. for example to check the breeds of an animal in comparison of the breed from his parents.

options

-h some help about the options.

- v print the version of the programm.
- p <project> project name same as defined in \$APIIS_HOME/etc/apiisrc.
- o <outfile> name of the outputfile. else printing on STDOUT.

6.9.12 WebForm.pl

ABSTRACT

WebForm.pl is a Web-Frontend on basis of yaform.pm. It reads *.frm-Files and create a Browser-Form for the input of datas.

PROGRAMMING STYLE

WebForm.pl is written in OO-Stile.

Notice

The communication of a browser with a database via HTML is very different to the communication via TK. That's why, not all features of Form.pl are implemented in WebForm.pl. For example getting information from the database by pushing the TAB-button.

Configuration

1.: apache runs under the user-id "wwwrun" (suse) , "www-data" (debian) or something like that. This user-id needs read-rights (group or other) for apiis/index and the project-directory and execute-rights for apiis/bin/WebForm.pl

2.: httpd.conf ("/etc/apache/" (debian) or "/etc/httpd/" (suse) needs this entry in "Section 3: Virtual Hosts": For using Internet each project needs a subdomain. There will be defined directories or variables. Do define subdomains a entry in /etc/httpd/httpd.conf for each subdomain is necessary.

```
#you must replace /home/... with your current pfad.
#####
<VirtualHost ref_breedprg.apiis.org>
    ServerName ref_breedprg.apiis.org
    ServerAlias ref_breedprg.apiis.org

    DocumentRoot /home/zwisss/devel/apiis/ref_breedprg
    ScriptAlias /cgi-bin/ /home/zwisss/devel/apiis/bin/
    Alias images /home/zwisss/devel/apiis/lib/images

    SetEnv APIIS_HOME /home/zwisss/devel/apiis
    SetEnv APIIS_LOCAL /home/zwisss/devel/apiis/ref_breedprg
    SetEnv HOME /home/zwisss

    ServerAdmin webmaster@www.apiis-sachsen.de>
</VirtualHost>
#####
```

3.: Add in `"/etc/hosts"`

```
127.0.0.2          ref_breedprg.localhost          ref_breedprg
```

If you define a `ServerAlias` in section `VirtualHost`, this entry isn't necessary.

4.: user-id `"wwwrun"` or `"www-data"` must exists in postgres

```
su root
su postgres
createuser wwwrun
```

Comment

Please note, all functions in `LO_*.pm` Objects and all global-variables must be expand to `main::`. For example `$main::dbh->rollback` or `main::CheckLO`, because, `WebForm.pl` is programmed in OO-Stile.

AUTHOR INFORMATION

Ulf Müller.

BUGS

Normally, the most features of `Form.pl` should be work, but not all possibilities were tested at time.

SEE ALSO

Apiis::Form::Base.pm, Apiis::Form::HTML.pm, Apiis::Form::Tk.pm

6.9.13

standalone wrapper for the `xml2model` subroutine

Chapter 7

Undocumented Subroutines

7.1 APIIS_HOME/lib/todo.js

7.2 APIIS_HOME/lib/menu_tpl1.js

7.3 APIIS_HOME/lib/apiis2css.js

7.4 APIIS_HOME/lib/form.js.org

7.5 APIIS_HOME/lib/apiis.xmllib

7.6 APIIS_HOME/lib/hierFrames.js

7.7 APIIS_HOME/lib/general.js

7.8 APIIS_HOME/lib/example_menu.js

7.9 APIIS_HOME/lib/formajax.js_neu

7.10 APIIS_HOME/lib/form_ulib.pm

1. GetNextDBID
2. Next_seq_DBID
3. SQL
4. call_LO
5. concat
6. concat2
7. div

8. existDBID
9. fetchlist
10. fromDB
11. fromField
12. getDBID
13. getDB_ID
14. getNextDBID
15. getNextDB_ID
16. getfile
17. invoke
18. login
19. multi
20. selFromDB
21. split
22. sub
23. sum
24. sysSQL

7.11 APIIS_HOME/lib/navigation.js

7.12 APIIS_HOME/lib/yaform.pm

1. Ini2ModelHash
2. Post
3. TkYAF
4. allesklar
5. block
6. browseDB
7. browseList
8. buttonBar
9. clearForm
10. commitData

11. coordinate
12. error
13. exeCommand
14. exeUserCommand
15. exitForm
16. fieldStatusWin
17. flash_background
18. focusIn
19. focusOut
20. function
21. getKey
22. getwidgetoption
23. initFORM
24. initFormHash
25. initModelHash
26. insert
27. motion
28. move
29. query
30. readWidget
31. recurs
32. resetBG COLOR
33. resetBalloons
34. setBalloon
35. setFirstButton
36. setLeftButton
37. setMaxButton
38. setNewButton
39. setRightButton
40. setform

- 41. setwidgetoption
- 42. startup
- 43. status
- 44. table_field_tableRelation
- 45. update
- 46. upin
- 47. warnwin
- 48. xcoord
- 49. ycoord

7.13 APIIS_HOME/lib/apiis2javascript.js

7.14 APIIS_HOME/lib/ref_breedprg_alib.pm

- 1. GetModelName
- 2. align
- 3. align2
- 4. align3
- 5. alpha
- 6. cntanc
- 7. cntanct3
- 8. commonpart
- 9. complete
- 10. complete2
- 11. complete3
- 12. completeness
- 13. create_repl_hash
- 14. eintrag
- 15. genepart
- 16. generations
- 17. get_animal_id
- 18. get_code_id

19. `get_db_code_db_unit`
20. `get_db_from_rowid`
21. `get_db_unit`
22. `get_dup_animal`
23. `get_ext_animal`
24. `get_ext_code`
25. `get_ext_val`
26. `get_id_hash_unit`
27. `get_index`
28. `get_last_val`
29. `get_next_val`
30. `get_string`
31. `get_unit_id`
32. `get_value`
33. `getdate`
34. `inbreed`
35. `insert_code_if_new`
36. `meuw`
37. `minusstr`
38. `pedanalys`
39. `plusstr`
40. `same_gene`
41. `sanitize`
42. `set_next_val`
43. `sminus`
44. `splus`
45. `test`
46. `testbd`
47. `testbd2`
48. `testgen`

- 49. testloop
- 50. testloop_array
- 51. testped
- 52. transfer_to_animal

7.15 **APIIS_HOME/lib/formajax.js**

7.16 **APIIS_HOME/lib/navigation.js**

7.17 **APIIS_HOME/lib/formneu.js**

7.18 **APIIS_HOME/lib/apiis2css.js**

7.19 **APIIS_HOME/lib/json.js**

7.20 **APIIS_HOME/lib/formnavigation.js**

7.21 **APIIS_HOME/lib/formajax.js.old**

7.22 **APIIS_HOME/lib/alt.js**

7.23 **APIIS_HOME/lib/Apiis/Load1.pm.org**

- 1. CollectCodesUnits
- 2. CreateRecordForPrint
- 3. HandleSourceFileInfos
- 4. PrintResults
- 5. PrintUnitsCodesChg
- 6. WriteLog

7.24 **APIIS_HOME/lib/Apiis/DataBase/Init.pm**

- 1. bindtypes
- 2. connect
- 3. connected_sys
- 4. connected_user
- 5. datatypes
- 6. disconnect

7. new
8. rollback
9. sys_commit
10. sys_dbh
11. sys_rollback
12. user_commit
13. user_dbh
14. user_rollback

7.25 APIIS_HOME/lib/Apiis/DataBase/Record.pm

1. RunTrigger
2. action
3. auth
4. check_level
5. check_record
6. decode_column
7. decode_record
8. decoded
9. delete
10. encode_column
11. encode_record
12. encoded
13. expect_columns
14. expect_rows
15. fetch
16. fk_table
17. indexes
18. insert
19. max_check_level
20. mirror_differs
21. mirror_extdata

- 22. mirror_intdata
- 23. mirror_record
- 24. mirrored
- 25. modify_record
- 26. pk_ext_cols
- 27. pk_ref_col
- 28. postdelete_triggers
- 29. postinsert_triggers
- 30. postupdate_triggers
- 31. predelete_triggers
- 32. preinsert_triggers
- 33. preupdate_triggers
- 34. print_loc
- 35. resolve_fk
- 36. resolve_pk
- 37. sequences
- 38. struct_type
- 39. tablename
- 40. triggeraction
- 41. type
- 42. update
- 43. value
- 44. values

7.26 APIIS_HOME/lib/Apiis/DataBase/User.pm

- 1. methods
- 2. sprint
- 1. read_state

7.27 APIIS_HOME/lib/Apiis/DataBase/Record.pm_

1. RunTrigger
2. action
3. auth
4. check_level
5. check_record
6. decode_column
7. decode_record
8. decoded
9. delete
10. encode_column
11. encode_record
12. encoded
13. expect_columns
14. expect_rows
15. fetch
16. fk_table
17. indexes
18. insert
19. max_check_level
20. mirror_differs
21. mirror_extdata
22. mirror_intdata
23. mirror_record
24. mirrored
25. modify_record
26. pk_ext_cols
27. pk_ref_col
28. postdelete_triggers
29. postinsert_triggers

- 30. postupdate_triggers
- 31. predelete_triggers
- 32. preinsert_triggers
- 33. preupdate_triggers
- 34. print_loc
- 35. resolve_fk
- 36. resolve_pk
- 37. sequences
- 38. struct_type
- 39. tablename
- 40. triggeraction
- 41. type
- 42. update
- 43. value
- 44. values

7.28 APIIS_HOME/lib/Apiis/DataBase/SQL/MakeSQL.pm

- 1. Cascaded_FK
- 2. HasFKRule
- 3. MakeSQL
- 4. resolve_concatenations

7.29 APIIS_HOME/lib/Apiis/DataBase/SQL/sql_flow_exam

- 1. ExecuteDelete
- 2. ExecuteInsert
- 3. ExecuteSelect
- 4. ExecuteUpdate
- 5. Proceed

7.30 APIIS_HOME/lib/Apiis/DataBase/SQL/DirectSQL.pm

- 1. sql
- 2. sys_sql

7.31 APIIS_HOME/lib/Apiis/DataBase/Record/Update.pm

1. SetOpeningDt
1. SetNow
1. SetUser

7.32 APIIS_HOME/lib/Apiis/DataBase/Record/Modify.pm

7.33 APIIS_HOME/lib/Apiis/DataBase/Record/Insert.pm

7.34 APIIS_HOME/lib/Apiis/DataBase/Record/Check.pm

7.35 APIIS_HOME/lib/Apiis/DataBase/Record/Trigger.pm

7.36 APIIS_HOME/lib/Apiis/DataBase/Record/Column.pm

1. ar_check
2. check
3. check_rules
4. datatype
5. db_column
6. decoded
7. default
8. description
9. encoded
10. ext_fields
11. ext_mirrored
12. extdata
13. foreignkey
14. form_type
15. int_mirrored
16. intdata
17. length
18. m_diff
19. m_diff_extdata

- 20. m_diff_intdata
- 21. m_extdata
- 22. m_intdata
- 23. mirror_extdata
- 24. mirror_intdata
- 25. modify
- 26. modify_rules
- 27. name
- 28. struct_type
- 29. tablename
- 30. tableobj
- 31. updated
 - 1. NotSpaceOnly
 - 1. ReservedStrings
 - 1. List
 - 1. NotNull

7.37 APIIS_HOME/lib/Apiis/DataBase/Record/Check/LastA

- 1. DateDiff
- 1. ForeignKey
- 1. IsEqual

7.38 APIIS_HOME/lib/Apiis/DataBase/Record/Trigger.pm

7.39 APIIS_HOME/lib/Apiis/DataBase/Record/Delete.pm

- 1. SetNow
- 1. UpperCase
- 1. ConvertBool
- 1. LowerCase
- 1. DotToColon
- 1. CommaToDot
- 1. SetUser

7.40 APIIS_HOME/lib/Apiis/Init.pm

1. DESTROY
2. check_date_conf
3. check_time_conf
4. date_order
5. date_parts
6. disconnect_project
7. extdate2iso
8. exttime2iso
9. filelog_priority
10. iso2extdate
11. iso2exttime
12. language
13. pid
14. servername
15. time_order
16. time_parts
17. use_sql_logging
18. use_syslog

7.41 APIIS_HOME/lib/Apiis/Report/Init.pm

1. GetAllElementsXML
2. GetAllParameter
3. GetAllSubReports
4. GetElementsXML
5. Methods
6. start

7.42 APIIS_HOME/lib/Apiis/Report/InitXML.pm

1. MergeAttributes
2. new

7.43 APIIS_HOME/lib/Apiis/Report/Base.pm

1. GetData
2. GetModelName
3. LinkModul
4. MakeReport
5. PrintFooter
6. PrintReport
7. Refresh
8. ResetFooterObjects
9. new

7.44 APIIS_HOME/lib/Apiis/Report/PDF.pm

1. PrintCell
2. PrintFooter
3. PrintHeader
4. PrintLongTable
5. PrintObjects
6. PrintReport
7. PrintRow
8. PrintTable
9. colortrans

7.45 APIIS_HOME/lib/Apiis/Report/HTML.pm

1. PrintCell
2. PrintHeader
3. PrintObjects
4. PrintRow
5. PrintTable

7.46 APIIS_HOME/lib/Apiis/GUI.pm

1. Apiis
2. GUI
3. GUIobj
4. GetData
5. GetParameter
6. LinkModul
7. MakeGUI
8. PrintAllCells
9. PrintFooter
10. PrintGUI
11. Query
12. Refresh
13. ResetFooterObjects
14. SetColumnBusy
15. SetParameter
16. Xml
17. new

7.47 APIIS_HOME/lib/Apiis/Load3.pm

1. AllSourceFileIds
2. CollectDataForPriorisation
3. GetCurrentTable
4. GetDbCodeUnit
5. GetRepeats
6. HandleSourceFileInfos
7. PrintResults
8. Priorisation
9. WriteFileWithInternalValues

7.48 APIIS_HOME/lib/Apiis/I18N/test.l10n**7.49 APIIS_HOME/lib/Apiis/Model.pm**

1. ar_check
2. basename
3. build_fk_struct
4. check
5. cols
6. datatype
7. db_driver
8. db_encoding
9. db_host
10. db_name
11. db_password
12. db_port
13. db_user
14. default
15. description
16. ext
17. form_type
18. fullname
19. has_fk_from
20. has_fk_struct
21. index
22. label
23. length
24. max_check_level
25. modify
26. path
27. sequence
28. triggers
29. views

7.50 APIIS_HOME/lib/Apiis/.Load3.pm.swp_**7.51 APIIS_HOME/lib/Apiis/Load.pm.org**

1. CheckID
2. CheckLogError
3. CheckUniqueAnimalNumber
4. CheckWrongAnimalNumber
5. CollectCodesUnits
6. CreateDBAndLoadData
7. CreateDbAnimalExtAnimal
8. CreateRecordForPrint
9. EncodeRecord
10. ExecuteStep
11. GetAttribute
12. GetCurrentTable
13. GetElement
14. GetPseudoAnimalNumbers
15. GetTempAttribute
16. HandleSourceFileInfos
17. LoadCodesUnitsOk
18. ParseXMLFile
19. PrepareCopyStatementsForLoading
20. PrintAnimalNumbers
21. PrintHashMemory
22. PrintResults
23. PrintTables
24. ReplaceCodesUnitsOldWithNew
25. SetForeignKey
26. TransformDataDate
27. WriteLog
28. start_tag

7.52 APIIS_HOME/lib/Apiis/Load1.pm

1. CollectCodesUnits
2. CreateRecordForPrint
3. HandleSourceFileInfos
4. PrintResults
5. PrintUnitsCodesChg
6. WriteLog

7.53 APIIS_HOME/lib/Apiis/Auth/AR_View.pm

1. create_views_where_clause
2. creates_table_view
3. entry_views

7.54 APIIS_HOME/lib/Apiis/Auth/AR_Auth.pm

1. check_columns
2. check_descriptors
3. check_sql_statement
4. check_system_tasks
5. check_tables
6. descriptor_fulfilled
7. methods
8. user_id

7.55 APIIS_HOME/lib/Apiis/Auth/test.AR_Auth

1. parameters
1. create_other_accounts
1. creates_public_views

(a) `update_session_lang`

(a) `get_arm_labels`

(a) `execute_user_sql`

7.85 `APIIS_HOME/lib/Apiis/Auth/ARww/lib/languages.`

(a) `show_documentation`

(a) `menu_order`

7.86 `APIIS_HOME/lib/Apiis/Auth/ARww/lib/HandleAJ.`

(a) `descriptors_define_sql`

(b) `main`

(a) `arm_main`

(a) `get`

i. new

7.114 APIIS_HOME/lib/Apiis/Errors.pm

i. action_values

ii. get_err_id

iii. inc_err_id

iv. severity_values

v. type_values

7.115 APIIS_HOME/lib/Apiis/Extjs/CHANGES_ext-3.1.0.html

7.116 APIIS_HOME/lib/Apiis/Extjs/gpl-3.0.txt

7.117 APIIS_HOME/lib/Apiis/Extjs/docs/output/tree

7.118 APIIS_HOME/lib/Apiis/Extjs/docs/resources/w

7.119 APIIS_HOME/lib/Apiis/Extjs/docs/resources/fa

7.120 APIIS_HOME/lib/Apiis/Extjs/docs/resources/s

7.121 APIIS_HOME/lib/Apiis/Extjs/docs/resources/p

7.122 APIIS_HOME/lib/Apiis/Extjs/docs/resources/p

7.123 APIIS_HOME/lib/Apiis/Extjs/docs/resources/w bg.gif

7.124 APIIS_HOME/lib/Apiis/Extjs/docs/resources/n collapsed.gif

7.125 APIIS_HOME/lib/Apiis/Extjs/docs/resources/t bg.gif

7.126 APIIS_HOME/lib/Apiis/Extjs/docs/resources/e members.gif

7.127 APIIS_HOME/lib/Apiis/Extjs/docs/resources/n expanded.gif

Index

- [dmyj] plus any character which split the dates, 83
- auth, 79
- init (internal), 107
- loadnodes (internal), 108
- \$apiis-> check date conf (internal), 68
- \$apiis-> check time conf (internal), 68
- \$apiis
 - >DataBase, 11
 - >sys_sql, 11
 - >user_sql, 11
 - current user, 11
 - driver configuration, 11
 - public methods, 11
 - system user, 11
 - >Model, 9
 - public methods, 9
 - >_add_object, 9
 - >join_model, 9, 11
 - >version, 8
 - public methods, 8
- Apiis pm
 - Author, 58
 - DESCRIPTION, 58
 - initialize, 58
 - SUBROUTINES, 58
 - SYNOPSIS, 57
- Apiis.pm, 57
- Apiis::Auth::AccessControl – used by the runall.pl and accesscontrol.pl scripts to define user access rights
 - access rights, 83
 - assigns db policies, 83
 - assigns os policies, 83
 - assigns role, 83
 - AUTHOR, 85
 - change password, 85
 - check policies, 83
 - class columns, 84
 - col in classes, 84
 - creates access view, 83
 - creates db policies, 83
 - creates role, 83
 - creates schema, 83
 - creates user, 83
 - creates view, 84
 - creates views v , 85
 - del role, 84
 - del role from user, 84
 - del user, 83
 - DESCRIPTION, 82
 - drop views, 84
 - main columns, 84
 - public views, 84
 - query db, 84
 - revoke priv, 84
 - select db, 84
 - SUBROUTINES, 83
 - SYNOPSIS, 82
- Apiis::Auth::AccessControl – used by the runall.pl and accesscontrol.pl scripts to define user access rights, 82
- Apiis::Auth::AppAuth – object for provading data about user rights for the applications, 81
 - get policy ids (internal), 82
 - get user id (internal), 81
 - get user roles (internal), 81
- AUTHORS, 82
- check os action (public), 82
- DESCRIPTION, 81
- METHODS, 81
 - new (public), 81
 - os actions, 82
 - print os actions (public), 82
- SYNOPSIS, 81
- types of actions (public), 82
- Apiis::Auth::Role pm
 - AUTHORS, 81
 - db policy get policy for the database from db policy section, 81
 - DESCRIPTION, 80
 - description returns description of role, 81
 - long name returns long role name, 81
 - METHODS, 80

- misc bloquelements (public), 111
- new (public), 109
- query block order (public), 111
- return value (public), 112
- RunEvent (public), 111
- SetValue (public), 110
- SYNOPSIS, 108
- top (public), 111
- xmlfile (public), 109
- Apiis::Init – Basic initialisation object for the complete APIIS structure, 58
 - (), 60
 - add obj (internal), 60
 - get user from os (internal), 59
 - init (internal), 59
 - join database (internal), 66
 - join user (internal), 63
 - \$apiis->[language] (public), 61
 - AUTOLOAD (internal), 67
 - check status (public), 64
 - debug (public), 63
 - del error (public), 65
 - del errors (public), 65
 - DESCRIPTION, 58
 - error (public), 65
 - errors (public), 65
 - exists auth (public), 67
 - exists database (public), 67
 - exists model (public), 66
 - exists user (public), 63
 - formpath (public), 59
 - get db conf (mainly internal), 67
 - INTERNAL METHODS, 59
 - join database (public), 66
 - join model (public), 66
 - l10n import (public), 59
 - l10n init (public), 59
 - localtime (public), 65
 - log (public), 64
 - log priority (public), 63
 - new (mostly internal), 59
 - now (public), 65
 - project (public), 59
 - projects (public), 59
 - PUBLIC METHODS, 60
 - running check integrity (public), 64
 - status (public), 64
 - substitute env (internal), 63
 - SYNOPSIS, 58
 - syslog priority/filelog priority (public), 63
 - today (public), 65
 - use filelog/use syslog/use sql logging (public), 63
- Apiis::Init::Config mainly ready apiisrc config files, 67
 - get db conf (internal), 68
 - import apiisrc (internal), 67
 - import apiisrc local (internal), 67
 - import user apiisrc (internal), 67
 - xml2model (internal), 68
 - DESCRIPTION, 67
 - METHODS, 67
- Apiis::Misc – Provides some usefull subroutines mainly for compatibility reasons
 - Decode Date NativeRDBMS, 77
 - DESCRIPTION, 76
 - elapsed, 78
 - Error, 77
 - file2variable, 78
 - find pod path, 78
 - Info, 77
 - is true, 78
 - LocalToRawDate, 77
 - MaskForLatex, 77
 - mimetype of, 79
 - mychomp, 77
 - RawToLocalDate, 77
 - show progress, 77
 - Subroutines, 77
 - SYNOPSIS, 76
- Apiis::Misc – Provides some usefull subroutines, mainly for compatibility reasons, 76
- Apiis::Model – methods to access the model file data via the \$apiis structure, 70
- Apiis::Model – methods to access the model file data via the apiis structure
 - init (internal), 70
- Apiis::Model::TableObj – internal package to provide a table object with methods to access a single table and its columns, 71

- check_level (public, read/write), 71
- DESCRIPTION, 70
- METHODS, 70
- new (mostly internal), 70
- SYNOPSIS, 70
- table (public, readonly), 70
- tables (public, readonly), 70
- Apiis::Model::ColumnObj – internal package to provide a column object with methods to access a single column of a table
 - DESCRIPTION, 73
 - foreignkey (public, readonly), 73
 - METHODS, 73
 - SYNOPSIS, 73
- Apiis::Model::TableObj – internal package to provide a table object with methods to access a single table and its columns
 - \$table obj->triggers(\$triggertype) (public, readonly), 72
- Apiis::Model::ColumnObj – internal package to provide a column object with methods to access a single column of a table, 73
- column (public, readonly), 71
- columns/cols (public, readonly), 71
- METHODS, 71
- name (public, readonly), 71
- new (mostly internal), 71
- primarykey (public, readonly), 72
- struct type (public, readonly), 71
- SYNOPSIS, 71
- APIIS_HOME/bin
 - .Rhistry, 158
 - access_control.pl
 - parameters, 158
 - access_rights_ar_batch.pl, 158
 - access_rights_manager.pl
 - parameters, 158
 - access_rights_update.pl
 - parameters, 158
 - add_gen_didier, 158
 - add_gen_didier-32bit, 158
 - add_gen_didier-64bit, 158
 - add_gen_didier.f90, 158
 - add_gen_didier.incl, 158
 - add_gen_didier_ulf, 158
 - Add_gen_didierReport.pl, 158
 - initialize, 158
 - round, 158
 - round2, 158
 - additive.o, 158
 - agr-extract_files, 158
 - usage, 158
 - agr-extract_files.ulf, 158
 - usage, 158
 - agr-run_parallel, 158
 - usage, 158
 - apiis-test-dependencies, 158
 - debian_inst, 158
 - resolve_dep, 158
 - test_dep, 158
 - apiish, 158
 - GetEntries, 158
 - RunReport, 158
 - RunReport_new, 158
 - check_pedi_loop, 158
 - draw_loop, 158
 - find_parents, 158
 - show_progress, 158
 - cleanup_incoming.sh, 158
 - Create_inbreeding_tables.pl, 158
 - counttable, 158
 - deltables, 158
 - get_ped_mem, 158
 - initialize, 158
 - report, 158
 - round0, 158
 - round3, 158
 - Create_Population_tables.pl, 158
 - usage, 158
 - CreatePediStack.pl, 158
 - chkbreed, 158
 - chkentry, 158
 - chkIID, 158
 - chksex, 158
 - get_ped_mem, 158
 - localtime, 158
 - test, 158
 - testloop, 158
 - cvs2cl, 158
 - add_file_entry, 158
 - add_fileentry, 158
 - branch, 158
 - branch_numbers, 158
 - clear_file, 158

- clear_msg, 158
- common_path_prefix, 158
- debug, 158
- delta_check, 158
- derive_changelog, 158
- dir_key, 158
- distributed, 158
- escape, 158
- fdate, 158
- fdatetime, 158
- filename, 158
- files, 158
- format_body, 158
- ftime, 158
- grand_poobah, 158
- header_line, 158
- last_line_len, 158
- lines, 158
- maybe_grab_accumulation_date, 158
- maybe_read_user_map_file, 158
- min, 158
- mywrap, 158
- new, 158
- output_changelog, 158
- output_entry, 158
- output_footer, 158
- output_header, 158
- output_tagdate, 158
- parse_date_author_and_state, 158
- parse_options, 158
- preprocess_msg_text, 158
- pretty_file_list, 158
- read_branches, 158
- read_changelog, 158
- read_date_author_and_state, 158
- read_file_path, 158
- read_revision, 158
- read_symbolic_name, 158
- revision, 158
- roots, 158
- run_ext, 158
- slurp_file, 158
- state, 158
- tags, 158
- tags_exists, 158
- time, 158
- usage, 158
- version, 158
- wday, 158
- weekday_en, 158
- wrap_log_entry, 158
- cvstat, 158
- datediff, 158
 - date_diff, 158
 - normalize_date, 158
 - Normalize_Delta_YMDHMS, 158
 - output_as_hours, 158
 - output_as_iso, 158
- dbf2txt, 158
- extract_for_blupMEM.pl, 158
 - get_ped_mem, 158
- file2inspool.pl, 158
 - usage, 158
- file_stats
 - myfind, 158
- FormDesigner
 - setFont, 158
- gnu_comp.txt, 158
- GUIN, 158
- handle_pedi_file, 158
 - check_date_conf, 158
 - check_encoding, 158
 - check_mime_type, 158
 - extdate2iso, 158
 - fill_ped, 158
- hashtable, 158
 - usage, 158
- inbreeding_report, 158
 - loginbreeding, 158
 - round, 158
 - round1, 158
 - round2, 158
 - round4, 158
 - usage, 158
- InbreedingReport.pl, 158
 - usage, 158
- is_valid_email, 158
- load_db_from_INSPPOOL, 158
 - usage, 158
- load_stat, 158
 - ascii, 158
 - latex, 158
 - usage, 158
- LoadProject.sh, 158
- log_of_Inbreeding.pm, 158
 - round, 158
 - round1, 158

- round2, 158
- round4, 158
- usage, 158
- MGUI, 158
- mk_rand_string, 158
- pw_generate, 158
- mkdescr, 158
- print_header, 158
- usage, 158
- mkl10n
- myfind, 158
- mkLOWform, 158
- mkxmlforms, 158
- CreateField, 158
- GetExternalValues, 158
- WriteDataSource, 158
- WriteField, 158
- WriteLabel, 158
- WriteTitle, 158
- model2xml.pl, 158
- ne_r_sloop_log_inbreeding.pl, 158
- initialize, 158
- round, 158
- round4, 158
- nodemanagement.pl
- get_classes, 158
- ped_completeness.pm, 158
- ped_completeness, 158
- traverse_pedigree, 158
- pedicomp.pl, 158
- get_ped_mem, 158
- initialize, 158
- round3, 158
- pedigree_loops.pl, 158
- get_ped, 158
- initialize, 158
- test, 158
- PopulationReport.pl, 158
- initialize, 158
- new_tabel, 158
- round, 158
- process_uploads.sh, 158
- READMEAdd_gen_didierReport.txt, 158
- READMEInbreedingReport.txt, 158
- READMEPopulationReport.txt, 158
- run_popreport, 158
- run_popreport_file, 158
- runexample.sh, 158
- selection.pl, 158
- Get_Selection_IDs_Hash, 158
- numerically, 158
- round, 158
- server.pl
- gotdata, 158
- show_rules, 158
- test.JSON, 158
- test.LO, 158
- test.Menu, 158
- test.pl, 158
- texfiles, 158
- upload.pl, 158
- WebForm.pl, 158
- browseList, 158
- commitData, 158
- exeUserCommand, 158
- getKey, 158
- status, 158
- xml2model.pl, 158
- APIIS_HOME/lib
- ajax.js, 158
- alt.js, 142
- alt.js , 158
- apiis.xmllib, 137
- apiis2css.js, 137
- apiis2css.js , 142
- apiis2javascript.js, 140
- apiis2javascript.js , 158
- example_menu.js, 137
- form.js, 158
- form.js.org, 137
- form_ulib.pm, 137
- call_LO, 137
- concat, 137
- concat2, 137
- div, 137
- existDBID, 138
- fetchlist, 138
- fromDB, 138
- fromField, 138
- getDB_ID, 138
- getDBID, 138
- getfile, 138
- getNextDB_ID, 138
- GetNextDBID, 137
- getNextDBID, 138
- invoke, 138
- login, 138
- multi, 138
- Next_seq_DBID, 137

- selFromDB, 138
- split, 138
- SQL, 137
- sub, 138
- sum, 138
- sysSQL, 138
- formajax.js, 142
- formajax.js.old, 142
- formajax.js_neu, 137
- formnavigation.js, 142
- formneu.js, 142
- formneu.js , 158
- general.js, 137
- general.js , 158
- hierArrays.js, 158
- hierFrames.js, 137
- json.js, 142
- menu.js, 158
- menu_items.js.org, 158
- menu_tpl1.js, 137
- myWriter.pm, 158
 - addPrefix, 158
 - ancestor, 158
 - characters, 158
 - comment, 158
 - current_element, 158
 - dataElement, 158
 - doctype, 158
 - emptyTag, 158
 - end, 158
 - endTag, 158
 - getDataIndent, 158
 - getDataMode, 158
 - getOutput, 158
 - in_element, 158
 - new, 158
 - pi, 158
 - removePrefix, 158
 - setDataIndent, 158
 - setDataMode, 158
 - setOutput, 158
 - startTag, 158
 - within_element, 158
 - xmlDecl, 158
- navigation.js, 142
- navigation.js , 138
- Popreport.pm, 158
 - db_clean_birthdates, 158
 - decode_animal, 158
 - draw_loop, 158
 - test_dates, 158
 - test_loop, 158
 - test_parent_is_known, 158
 - test_sex, 158
 - test_sire_eq_dam, 158
- ref_breedprg_alib.pm, 140
 - align, 140
 - align2, 140
 - align3, 140
 - alpha, 140
 - cntanc, 140
 - cntanct3, 140
 - commonpart, 140
 - complete, 140
 - complete2, 140
 - complete3, 140
 - completeness, 140
 - create_repl_hash, 140
 - eintrag, 140
 - genepart, 140
 - generations, 140
 - get_animal_id, 140
 - get_code_id, 140
 - get_db_code_db_unit, 141
 - get_db_from_rowid, 141
 - get_db_unit, 141
 - get_dup_animal, 141
 - get_ext_animal, 141
 - get_ext_code, 141
 - get_ext_val, 141
 - get_id_hash_unit, 141
 - get_index, 141
 - get_last_val, 141
 - get_next_val, 141
 - get_string, 141
 - get_unit_id, 141
 - get_value, 141
 - getdate, 141
 - GetModelName, 140
 - inbreed, 141
 - insert_code_if_new, 141
 - meuw, 141
 - minusstr, 141
 - pedanalys, 141
 - plusstr, 141
 - same_gene, 141
 - sanitize, 141
 - set_next_val, 141
 - sminus, 141
 - splus, 141

- test, 141
- testbd, 141
- testbd2, 141
- testgen, 141
- testloop, 142
- testloop_array, 142
- testped, 142
- transfer_to_animal, 142
- todo.js, 137
- XMLConversion.pm
 - end, 158
- yaform.pm, 138
 - allesklar, 138
 - block, 138
 - browseDB, 138
 - browseList, 138
 - buttonBar, 138
 - clearForm, 138
 - commitData, 138
 - coordinate, 139
 - error, 139
 - exeCommand, 139
 - exeUserCommand, 139
 - exitForm, 139
 - fieldStatusWin, 139
 - flash_background, 139
 - focusIn, 139
 - focusOut, 139
 - function, 139
 - getKey, 139
 - getwidgetoption, 139
 - Ini2ModelHash, 138
 - initFORM, 139
 - initFormHash, 139
 - initModelHash, 139
 - insert, 139
 - motion, 139
 - move, 139
 - Post, 138
 - query, 139
 - readWidget, 139
 - recurs, 139
 - resetBalloons, 139
 - resetBGCOLOR, 139
 - setBalloon, 139
 - setFirstButton, 139
 - setform, 139
 - setLeftButton, 139
 - setMaxButton, 139
 - setNewButton, 139
 - setRightButton, 139
 - setwidgetoption, 140
 - startup, 140
 - status, 140
 - table_field_tableRelation, 140
 - TkYAF, 138
 - update, 140
 - upin, 140
 - warnwin, 140
 - xcoord, 140
 - ycoord, 140
- APIIS_HOME/lib/Apiis
 - .Load3.pm.swp_, 153
 - CheckFile.pm
 - FindFile, 158
 - Errors.pm, 158
 - action_values, 158
 - get_err_id, 158
 - inc_err_id, 158
 - severity_values, 158
 - type_values, 158
 - GUI.pm, 151
 - Apiis, 151
 - GetData, 151
 - GetParameter, 151
 - GUI, 151
 - GUObj, 151
 - LinkModul, 151
 - MakeGUI, 151
 - new, 151
 - PrintAllCells, 151
 - PrintFooter, 151
 - PrintGUI, 151
 - Query, 151
 - Refresh, 151
 - ResetFooterObjects, 151
 - SetColumnBusy, 151
 - SetParameter, 151
 - Xml, 151
 - Init.pm, 149
 - check_date_conf, 149
 - check_time_conf, 149
 - date_order, 149
 - date_parts, 149
 - DESTROY, 149
 - disconnect_project, 149
 - extdate2iso, 149
 - exttime2iso, 149
 - filelog_priority, 149
 - iso2extdate, 149

- iso2exttime, 149
- language, 149
- pid, 149
- servername, 149
- time_order, 149
- time_parts, 149
- use_sql_logging, 149
- use_syslog, 149
- Load.pm, 158
 - CheckID, 158
 - CheckLogError, 158
 - CheckUniqueAnimalNumber, 158
 - CheckWrongAnimalNumber, 158
 - CollectCodesUnits, 158
 - CreateDBAndLoadData, 158
 - CreateDbAnimalExtAnimal, 158
 - CreateRecordForPrint, 158
 - EncodeRecord, 158
 - ExecuteStep, 158
 - GetAttribute, 153
 - GetCurrentTable, 153
 - GetElement, 153
 - GetPseudoAnimalNumbers, 153
 - GetTempAttribute, 153
 - HandleSourceFileInfos, 153
 - LoadCodesUnitsOk, 153
 - ParseXMLFile, 153
 - PrepareCopyStatementsForLoading, 153
 - PrintAnimalNumbers, 153
 - PrintHashMemory, 153
 - PrintResults, 153
 - PrintTables, 153
 - ReplaceCodesUnitsOldWithNew, 153
 - SetForeignKey, 153
 - start_tag, 153
 - TransformDataDate, 153
 - WriteLog, 153
- Load1.pm, 154
 - CollectCodesUnits, 154
 - CreateRecordForPrint, 154
 - HandleSourceFileInfos, 154
 - PrintResults, 154
 - PrintUnitsCodesChg, 154
 - WriteLog, 154
- Load1.pm.org, 142
 - CollectCodesUnits, 142
 - CreateRecordForPrint, 142
 - HandleSourceFileInfos, 142
 - PrintResults, 142
 - PrintUnitsCodesChg, 142
 - WriteLog, 142
- Load2.pm, 158
 - AppendNewNumber, 158
 - AppendNumber, 158
 - CheckAnimalNumber, 158
 - CheckStatus, 158
 - CreateRecordForPrint, 158
 - HandleSourceFileInfos, 158
 - MergeDbAnimal, 158
 - PrintAnimalNumbers, 158
 - PrintResults, 158
 - ReplaceCodesUnitsOldWithNew, 158
 - SetForeignKey, 158
- Load3.pm, 151
 - AllSourceFileIds, 151
 - CollectDataForPriorisation, 151
- Load.pm.org, 153
 - CheckID, 153
 - CheckLogError, 153
 - CheckUniqueAnimalNumber, 153
 - CheckWrongAnimalNumber, 153
 - CollectCodesUnits, 153
 - CreateDBAndLoadData, 153
 - CreateDbAnimalExtAnimal, 153
 - CreateRecordForPrint, 153
 - EncodeRecord, 153
 - ExecuteStep, 153

- GetCurrentTable, 151
- GetDbCodeUnit, 151
- GetRepeats, 151
- HandleSourceFileInfos, 151
- PrintResults, 151
- Priorisation, 151
- WriteFileWithInternalValues, 151
- Load4.pm
 - CreateDBAndLoadData, 158
- Misc.pm
 - MaskForLatex, 158
- Model.pm, 152
 - ar_check, 152
 - basename, 152
 - build_fk_struct, 152
 - check, 152
 - cols, 152
 - datatype, 152
 - db_driver, 152
 - db_encoding, 152
 - db_host, 152
 - db_name, 152
 - db_password, 152
 - db_port, 152
 - db_user, 152
 - default, 152
 - description, 152
 - ext, 152
 - form_type, 152
 - fullname, 152
 - has_fk_from, 152
 - has_fk_struct, 152
 - index, 152
 - label, 152
 - length, 152
 - max_check_level, 152
 - modify, 152
 - path, 152
 - sequence, 152
 - triggers, 152
 - views, 152
- APIIS_HOME/lib/Apiis/Auth
 - AccessControl.pm
 - creates_public_views, 154
 - AR_Auth.pm, 154
 - check_columns, 154
 - check_descriptors, 154
 - check_sql_statement, 154
 - check_system_tasks, 154
 - check_tables, 154
 - descriptor_fulfilled, 154
 - methods, 154
 - user_id, 154
 - AR_Init.pm
 - create_other_accounts, 154
 - AR_View.pm, 154
 - create_views_where_clause, 154
 - creates_table_view, 154
 - entry_views, 154
 - Role.pm
 - new, 158
 - test.AR_Auth, 154
 - parameters, 154
- APIIS_HOME/lib/Apiis/Auth/AR_{ww}/cgi-bin
 - set_lang.cgi
 - update_session_lang, 156
- APIIS_HOME/lib/Apiis/Auth/AR_{ww}/lib
 - ARMFormMenu.pm
 - get, 156
 - ARMGeneral.pm
 - show_documentation, 156
 - ARMLabels.pm
 - get_arm_labels, 156
 - ARMMain.pm
 - arm_main, 156
 - ARMReadDB.pm
 - execute_user_sql, 156
 - HandleAJAX.pm, 156
 - descriptors_define_sql, 156
 - main, 156
 - languages.dat, 156
 - WebMenu.pm
 - menu_order, 156
- APIIS_HOME/lib/Apiis/DataBase
 - Init.pm, 142
 - bindtypes, 142
 - connect, 142
 - connected_sys, 142
 - connected_user, 142
 - datatypes, 142
 - disconnect, 142
 - new, 143
 - rollback, 143
 - sys_commit, 143
 - sys_dbh, 143
 - sys_rollback, 143
 - user_commit, 143
 - user_dbh, 143
 - user_rollback, 143

- Record.pm, 143
 - action, 143
 - auth, 143
 - check_level, 143
 - check_record, 143
 - decode_column, 143
 - decode_record, 143
 - decoded, 143
 - delete, 143
 - encode_column, 143
 - encode_record, 143
 - encoded, 143
 - expect_columns, 143
 - expect_rows, 143
 - fetch, 143
 - fk_table, 143
 - indexes, 143
 - insert, 143
 - max_check_level, 143
 - mirror_differs, 143
 - mirror_extdata, 143
 - mirror_intdata, 144
 - mirror_record, 144
 - mirrored, 144
 - modify_record, 144
 - pk_ext_cols, 144
 - pk_ref_col, 144
 - postdelete_triggers, 144
 - postinsert_triggers, 144
 - postupdate_triggers, 144
 - predelete_triggers, 144
 - preinsert_triggers, 144
 - preupdate_triggers, 144
 - print_loc, 144
 - resolve_fk, 144
 - resolve_pk, 144
 - RunTrigger, 143
 - sequences, 144
 - struct_type, 144
 - tablename, 144
 - triggeraction, 144
 - type, 144
 - update, 144
 - value, 144
 - values, 144
- Record.pm_, 145
 - action, 145
 - auth, 145
 - check_level, 145
 - check_record, 145
 - decode_column, 145
 - decode_record, 145
 - decoded, 145
 - delete, 145
 - encode_column, 145
 - encode_record, 145
 - encoded, 145
 - expect_columns, 145
 - expect_rows, 145
 - fetch, 145
 - fk_table, 145
 - indexes, 145
 - insert, 145
 - max_check_level, 145
 - mirror_differs, 145
 - mirror_extdata, 145
 - mirror_intdata, 145
 - mirror_record, 145
 - mirrored, 145
 - modify_record, 145
 - pk_ext_cols, 145
 - pk_ref_col, 145
 - postdelete_triggers, 145
 - postinsert_triggers, 145
 - postupdate_triggers, 146
 - predelete_triggers, 146
 - preinsert_triggers, 146
 - preupdate_triggers, 146
 - print_loc, 146
 - resolve_fk, 146
 - resolve_pk, 146
 - RunTrigger, 145
 - sequences, 146
 - struct_type, 146
 - tablename, 146
 - triggeraction, 146
 - type, 146
 - update, 146
 - value, 146
 - values, 146
- User.pm, 144
 - methods, 144
 - sprint, 144
- APIIS_HOME/lib/Apiis/DataBase/Record
 - Check.pm, 147
 - Column.pm, 147
 - ar_check, 147
 - check, 147
 - check_rules, 147
 - datatype, 147

- db_column, 147
- decoded, 147
- default, 147
- description, 147
- encoded, 147
- ext_fields, 147
- ext_mirrored, 147
- extdata, 147
- foreignkey, 147
- form_type, 147
- int_mirrored, 147
- intdata, 147
- length, 147
- m_diff, 147
- m_diff_extdata, 147
- m_diff_intdata, 148
- m_extdata, 148
- m_intdata, 148
- mirror_extdata, 148
- mirror_intdata, 148
- modify, 148
- modify_rules, 148
- name, 148
- struct_type, 148
- tablename, 148
- tableobj, 148
- updated, 148
- Delete.pm, 148
- Insert.pm, 147
- Modify.pm, 147
- Trigger.pm, 147
- Trigger.pm_, 148
- Update.pm, 147
- APIIS_HOME/lib/Apiis/DataBase/Record/Check
- DateDiff.pm
 - DateDiff, 148
- ForeignKey.pm
 - ForeignKey, 148
- IsEqual.pm
 - IsEqual, 148
- LastAction.pm, 148
- List.pm
 - List, 148
- NotNull.pm
 - NotNull, 148
- NotSpaceOnly.pm
 - NotSpaceOnly, 148
- ReservedStrings.pm
 - ReservedStrings, 148
- APIIS_HOME/lib/Apiis/DataBase/Record/CommaToDot.pm
 - CommaToDot, 148
- ConvertBool.pm
 - ConvertBool, 148
- DotToColon.pm
 - DotToColon, 148
- LowerCase.pm
 - LowerCase, 148
- SetNow.pm
 - SetNow, 148
- SetUser.pm
 - SetUser, 148
- UpperCase.pm
 - UpperCase, 148
- APIIS_HOME/lib/Apiis/DataBase/Record/Trigger
- SetNow.pm
 - SetNow, 147
- SetOpeningDt.pm
 - SetOpeningDt, 147
- SetUser.pm
 - SetUser, 147
- APIIS_HOME/lib/Apiis/DataBase/SQL
- DirectSQL.pm, 146
 - sql, 146
 - sys_sql, 146
- MakeSQL.pm, 146
 - Cascaded_FK, 146
 - HasFKRule, 146
 - MakeSQL, 146
 - resolve_concatenations, 146
- sql_flow_example.pl, 146
 - ExecuteDelete, 146
 - ExecuteInsert, 146
 - ExecuteSelect, 146
 - ExecuteUpdate, 146
 - Proceed, 146
- APIIS_HOME/lib/Apiis/DataBase/Sync
- Node.pm
 - read_state, 144
- APIIS_HOME/lib/Apiis/Extjs
- CHANGES_ext-3.1.0.html, 158
- ext-all-debug.js, 158
- ext-all.js, 158
- ext.jsb2, 158
- gpl-3.0.txt, 158
- INCLUDE_ORDER.txt, 158
- license.txt, 158
- APIIS_HOME/lib/Apiis/Extjs/adapter/ext
- ext-base-debug-w-comments.js, 158
- ext-base-debug.js, 158
- ext-base-debug.js, 158

- ext-base.js, 158
- APIIS_HOME/lib/Apiis/Extjs/adapters/jquery/jquery.prettify.js, 158
- ext-jquery-adapter-debug.js, 158
- ext-jquery-adapter.js, 158
- APIIS_HOME/lib/Apiis/Extjs/adapters/prototype/prototype.js, 158
- ext-prototype-adapter-debug.js, 158
- ext-prototype-adapter.js, 158
- APIIS_HOME/lib/Apiis/Extjs/adapters/yui/yui-combo-more.html, 158
- ext-yui-adapter-debug.js, 158
- ext-yui-adapter.js, 158
- APIIS_HOME/lib/Apiis/Extjs/docs/output/Element.scroll-more.html, 158
- tree.js, 158
- APIIS_HOME/lib/Apiis/Extjs/docs/resources/Element.traversal-more.html, 158
- block-bg.gif, 158
- block-bottom.gif, 158
- block-top.gif, 158
- brick.png, 158
- collapse-all.gif, 158
- collapse-bg.gif, 158
- collapser.css, 158
- docs.css, 158
- docs.js, 158
- elbow-end.gif, 158
- expand-all.gif, 158
- expand-bg-over.gif, 158
- expand-bg.gif, 158
- expand-members.gif, 158
- ext-all.js, 158
- ext-base.js, 158
- favicon.ico, 158
- form.png, 158
- hd-bg.gif, 158
- hide-inherited.gif, 158
- icon-grid.gif, 158
- member-collapsed.gif, 158
- member-expanded.gif, 158
- member-hover.gif, 158
- pkg-closed.gif, 158
- pkg-open.gif, 158
- print.css, 158
- reset.css, 158
- style.css, 158
- TabCloseMenu.js, 158
- tree-bg.gif, 158
- wel-bg.gif, 158
- welcome-bg.gif, 158
- welcome.css, 158
- APIIS_HOME/lib/Apiis/Extjs/docs/resources/EventManager-more.html, 158
- ext-base-ajax.html, 158
- ext-base-anim-extra.html, 158
- ext-base-anim.html, 158
- ext-base-begin.html, 158
- ext-base-dom-more.html, 158
- ext-base-dom.html, 158
- ext-base-end.html, 158
- ext-base-event.html, 158
- ext-base-point.html, 158
- ext-base-region.html, 158
- ext-lang-af.html, 158
- ext-lang-bg.html, 158
- ext-lang-ca.html, 158
- ext-lang-cs.html, 158
- ext-lang-da.html, 158
- ext-lang-de.html, 158
- ext-lang-el_GR.html, 158
- ext-lang-en.html, 158
- ext-lang-en_GB.html, 158
- ext-lang-es.html, 158
- ext-lang-fa.html, 158
- ext-lang-fi.html, 158
- ext-lang-fr.html, 158
- ext-lang-fr_CA.html, 158
- ext-lang-gr.html, 158
- ext-lang-he.html, 158
- ext-lang-hr.html, 158
- ext-lang-hu.html, 158
- ext-lang-id.html, 158
- ext-lang-it.html, 158
- ext-lang-ja.html, 158
- ext-lang-ko.html, 158
- ext-lang-lt.html, 158
- ext-lang-lv.html, 158
- ext-lang-mk.html, 158
- ext-lang-nl.html, 158

- ext-lang-no_NB.html, 158
- ext-lang-no_NN.html, 158
- ext-lang-pl.html, 158
- ext-lang-pt.html, 158
- ext-lang-pt_BR.html, 158
- ext-lang-pt_PT.html, 158
- ext-lang-ro.html, 158
- ext-lang-ru.html, 158
- ext-lang-sk.html, 158
- ext-lang-sl.html, 158
- ext-lang-sr.html, 158
- ext-lang-sr_RS.html, 158
- ext-lang-sv_SE.html, 158
- ext-lang-th.html, 158
- ext-lang-tr.html, 158
- ext-lang-ukr.html, 158
- ext-lang-vn.html, 158
- ext-lang-zh_CN.html, 158
- ext-lang-zh_TW.html, 158
- Ext-more.html, 158
- jquery-bridge.html, 158
- Observable-more.html, 158
- prototype-bridge.html, 158
- Template-more.html, 158
- yui-bridge.html, 158
- APIIS_HOME/lib/Apiis/Extjs/examples
 - .DS_Store, 158
 - data.js, 158
 - init.js, 158
 - README.txt, 158
 - samples.css, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/buttonviewer
 - buttons.css, 158
 - buttons.js, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/charts
 - charts.js, 158
 - pie-chart.html, 158
 - pie-chart.js, 158
 - reload-chart.html, 158
 - reload-chart.js, 158
 - stacked-bar-chart.html, 158
 - stacked-bar-chart.js, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/combos
 - spotlight-example.js, 158
 - templates.js, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/dd
 - dnd_grid_to_formpanel.js, 158
 - dnd_grid_to_grid.js, 158
 - dragdropzones.js, 158
 - field-to-grid-dd.html, 158
 - field-to-grid-dd.js, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/debug
 - debug-console.html, 158
 - debug.png, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/desktop
 - credits.txt, 158
 - sample.js, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/desktop/css
 - desktop.css, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/desktop/js
 - App.js, 158
 - Desktop.js, 158
 - Module.js, 158
 - StartMenu.js, 158
 - TaskBar.js, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/direct
 - direct-form.js, 158
 - direct-form.php, 158
 - direct-tree.js, 158
 - direct-tree.php, 158
 - direct.js, 158
 - direct.php, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/direct/php
 - api.php, 158
 - config.php, 158
 - poll.php, 158
 - router.php, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/direct/php/classes
 - Profile.php, 158
 - TestAction.php, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/feed-
 - feed-proxy.php, 158
 - feed-viewer.css, 158
 - FeedGrid.js, 158
 - FeedPanel.js, 158
 - FeedViewer.js, 158
 - FeedWindow.js, 158
 - MainPanel.js, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/form
 - absform.js, 158
 - adv-vtypes.html, 158
 - adv-vtypes.js, 158
 - anchoring.js, 158
 - check-radio.html, 158
 - check-radio.js, 158
 - combos.css, 158
 - combos.js, 158
 - custom.js, 158
 - dynamic.js, 158

- file-upload.html, 158
- file-upload.js, 158
- file-upload.php, 158
- form-grid.html, 158
- form-grid.js, 158
- forms.css, 158
- forum-search.html, 158
- forum-search.js, 158
- states.js, 158
- vbox-form.html, 158
- vbox-form.js, 158
- xml-errors.xml, 158
- xml-form.html, 158
- xml-form.js, 158
- xml-form.xml, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/forum
 - cmp-bg.gif, 158
 - forum.css, 158
 - forum.js, 158
 - message.png, 158
 - message_edit.png, 158
 - messages.png, 158
 - preview.png, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/grid
 - array-grid.html, 158
 - array-grid.js, 158
 - binding-with-classes.html, 158
 - binding-with-classes.js, 158
 - binding.js, 158
 - buffer.js, 158
 - column-header-group.js, 158
 - edit-grid.html, 158
 - edit-grid.js, 158
 - from-markup.html, 158
 - from-markup.js, 158
 - gen-names.js, 158
 - grid-examples.css, 158
 - grid-plugins.html, 158
 - grid-plugins.js, 158
 - grouping.css, 158
 - grouping.js, 158
 - locking-grid.html, 158
 - locking-grid.js, 158
 - paging.js, 158
 - progress-bar-pager.html, 158
 - progress-bar-pager.js, 158
 - property-grid.html, 158
 - property-grid.js, 158
 - row-editor-data.json, 158
 - row-editor.html, 158
 - row-editor.js, 158
 - sliding-pager.html, 158
 - sliding-pager.js, 158
 - totals-hybrid.html, 158
 - totals-hybrid.js, 158
 - totals-hybrid.json, 158
 - totals.js, 158
 - xml-grid.html, 158
 - xml-grid.js, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/grid-
 - filtering
 - grid-demo.sql, 158
 - grid-filter-local.html, 158
 - grid-filter-local.js, 158
 - grid-filter.json, 158
 - grid-filter.php, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/grouptabs
 - grouptabs.js, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/history
 - history.js, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/image-
 - organizer
 - img-org.css, 158
 - readme.txt, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/image-
 - organizer/imgorg
 - AlbumsPanel.js, 158
 - AlbumTree.js, 158
 - AlbumWin.js, 158
 - App.js, 158
 - Checkable.js, 158
 - DirectCombo.js, 158
 - ImageDv.js, 158
 - ImagePanel.js, 158
 - ImageThumbPanel.js, 158
 - MultiCombo.js, 158
 - TagWin.js, 158
 - UploadQueue.js, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/image-
 - organizer/php
 - api.php, 158
 - config.php, 158
 - poll.php, 158
 - router.php, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/image-
 - organizer/php/classes
 - Albums.php, 158
 - Images.php, 158
 - Tags.php, 158

- APIIS_HOME/lib/Apiis/Extjs/examples/images
 - organizer/php/sql
 - preview.png, 158
 - setup.php, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/image-box
 - organizer/SWFUpload
 - msg-box.html, 158
 - msg-box.js, 158
 - swfupload license.txt, 158
 - swfupload.js, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/image-multiselect
 - organizer/SWFUpload/Flash
 - multiselect-demo.html, 158
 - multiselect-demo.js, 158
 - swfupload.swf, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/image-organizer
 - organizer/SWFUpload/plugins
 - organizer.css, 158
 - organizer.js, 158
 - SWFObject License.txt, 158
- swfupload.cookies.js, 158
- swfupload.queue.js, 158
- swfupload.speed.js, 158
- swfupload.swfobject.js, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/panels
 - feed-viewer
 - feed-proxy.php, 158
 - feed-viewer.css, 158
 - FeedGrid.js, 158
 - FeedPanel.js, 158
 - FeedViewer.js, 158
 - FeedWindow.js, 158
 - MainPanel.js, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/panels-bubble
 - APIIS_HOME/lib/Apiis/Extjs/examples/panels/css
 - bubble.css, 158
 - APIIS_HOME/lib/Apiis/Extjs/examples/portal
 - portal.js, 158
 - sample-grid.js, 158
 - sample.css, 158
 - APIIS_HOME/lib/Apiis/Extjs/examples/resizable
 - basic.css, 158
 - basic.js, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/panels-restful
 - APIIS_HOME/lib/Apiis/Extjs/examples/restful
 - app.php, 158
 - restful-thumb.gif, 158
 - restful.css, 158
 - restful.js, 158
 - APIIS_HOME/lib/Apiis/Extjs/examples/restful/remote
 - init.php, 158
 - APIIS_HOME/lib/Apiis/Extjs/examples/restful/remote/app
 - users.php, 158
 - APIIS_HOME/lib/Apiis/Extjs/examples/restful/remote/app
 - user.php, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/panels-restful-lib
 - APIIS_HOME/lib/Apiis/Extjs/examples/restful/remote/lib
 - application_controller.php, 158
 - model.php, 158
 - request.php, 158
 - response.php, 158
 - session_db.php, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/shared
 - code-display.js, 158
 - examples.css, 158
 - examples.js, 158
 - info-bg.gif, 158
 - lib-bar.png, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/shared/extjs
 - actions.js, 158
 - list-items.gif, 158
 - menu-show.gif, 158
 - menus.css, 158

- App.js, 158
- extjs2.png, 158
- ft.png, 158
- site.js, 158
- topbar.png, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/shared/
 - extjs.css, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/shared/
 - arrow-down.gif, 158
 - arrow-up.gif, 158
 - silk.css, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/shared/
 - accept.png, 158
 - add.png, 158
 - application_go.png, 158
 - application_view_list.png, 158
 - book.png, 158
 - cog.png, 158
 - cog_edit.png, 158
 - connect.png, 158
 - control_rewind.png, 158
 - error.png, 158
 - feed_add.png, 158
 - feed_delete.png, 158
 - feed_error.png, 158
 - folder_go.png, 158
 - folder_wrench.png, 158
 - grid.png, 158
 - image_add.png, 158
 - information.png, 158
 - rss_go.png, 158
 - SILK.txt, 158
 - table_refresh.png, 158
 - user.png, 158
 - user_add.png, 158
 - user_comment.png, 158
 - user_delete.png, 158
 - user_edit.png, 158
 - user_female.png, 158
 - user_gray.png, 158
 - user_green.png, 158
 - user_orange.png, 158
 - user_red.png, 158
 - user_suit.png, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/shared/
 - acc-tree.gif, 158
 - border-layout.gif, 158
 - chart-pie.gif, 158
 - chart-reload.gif, 158
 - chart-stacked.gif, 158
 - combo-custom.gif, 158
 - data-view.gif, 158
 - dd-fieldtogrid.gif, 158
 - dd-gridtoformpanel.gif, 158
 - dd-gridtogrid.gif, 158
 - debug-console.gif, 158
 - form-indenttypes.gif, 158
 - form-check-radio.gif, 158
 - form-custom.gif, 158
 - form-dynamic.gif, 158
 - form-fieldsload.gif, 158
 - form-grid-binding.gif, 158
 - form-multiselect.gif, 158
 - form-spinner.gif, 158
 - form-vbox.gif, 158
 - form-xml.gif, 158
 - gmap-panel.gif, 158
 - grid-array.gif, 158
 - grid-buffer.gif, 158
 - grid-columngrouping.gif, 158
 - grid-data-binding.gif, 158
 - grid-edit.gif, 158
 - grid-filter.gif, 158
 - grid-from-markup.gif, 158
 - grid-grouping.gif, 158
 - grid-locking.gif, 158
 - grid-paging.gif, 158
 - grid-plugins.gif, 158
 - grid-property.gif, 158
 - grid-row-editor.gif, 158
 - grid-summary.gif, 158
 - grid-xml.gif, 158
 - group-tabs.gif, 158
 - layout-absolute.gif, 158
 - layout-accordion.gif, 158
 - layout-anchor.gif, 158
 - layout-browser.gif, 158
 - layout-column.gif, 158
 - layout-form.gif, 158
 - layout-table.gif, 158
 - layout-vbox.gif, 158
 - list-view.gif, 158
 - locale-dutch.gif, 158
 - locale-english.gif, 158
 - msg-box.gif, 158
 - panel-bubble.gif, 158
 - progress-bar-pager.gif, 158
 - restful-thumb.gif, 158
 - slider-pager.gif, 158

- statusbar-adv.gif, 158
- statusbar-demo.gif, 158
- tab-panel-scroller-menu.gif, 158
- tabs-adv.gif, 158
- toolbar-actions.gif, 158
- toolbar-button-groups.gif, 158
- toolbar-overflow.gif, 158
- tree-check.gif, 158
- tree-columns.gif, 158
- tree-reorder.gif, 158
- tree-two.gif, 158
- tree-xml-loader.gif, 158
- window-layout.gif, 158
- writer-thumb.gif, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/simply
 - widgets
 - ajax-tip.html, 158
 - editor.js, 158
 - progress-bar.css, 158
 - progress-bar.html, 158
 - progress-bar.js, 158
 - qtips.css, 158
 - qtips.js, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/slider
 - slider.css, 158
 - slider.js, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/spinner
 - spinner.js, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/statusbar
 - get-state.php, 158
 - README.txt, 158
 - save-state.php, 158
 - SessionProvider.js, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/statusbar-advanced
 - fake.php, 158
 - statusbar-advanced.html, 158
 - statusbar-advanced.js, 158
 - statusbar-demo.html, 158
 - statusbar-demo.js, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/tabs
 - ajax1.htm, 158
 - ajax2.htm, 158
 - tab-scroller-menu.css, 158
 - tab-scroller-menu.gif, 158
 - tab-scroller-menu.html, 158
 - tab-scroller-menu.js, 158
 - tabs-adv.html, 158
 - tabs-adv.js, 158
 - tabs-example.css, 158
 - tabs-example.js, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/tasks
 - classes.js, 158
 - gears.js, 158
 - tasks.css, 158
 - tasks.js, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/tasks/db
 - ext-air-db.js, 158
 - ext-db.js, 158
 - ext-gears-db.js, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/test-case-2b
 - .DS_Store, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/test-case-2b/resources
 - bubble-cs-fleedd.gif, 158
 - bubble-fleedd.css, 158
 - bubble-lr.gif, 158
 - bubble-tb.gif, 158
 - form-configs.js, 158
 - grid-configs.js, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/toolbar
 - overflow.js, 158
 - toolbars.css, 158
 - toolbars.js, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/tree
 - aria-tree.html, 158
 - aria-tree.js, 158
 - ARIA.js, 158
 - center-bg.gif, 158
 - check-nodes.json, 158
 - check-tree.html, 158
 - check-tree.js, 158
 - column-data.json, 158
 - column-tree.css, 158
 - column-tree.html, 158
 - column-tree.js, 158
 - get-nodes.php, 158
 - reorder.js, 158
 - save-dep.php, 158
 - two-trees.html, 158
 - two-trees.js, 158
 - xml-tree-data.xml, 158
 - xml-tree-loader.css, 158
 - xml-tree-loader.html, 158
 - xml-tree-loader.js, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/treegrid
 - tree-grid.js, 158
 - treegrid-data.json, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/ux
 - BufferView.js, 158

- CenterLayout.js, 158
- CheckColumn.js, 158
- ColumnHeaderGroup.js, 158
- ColumnNodeUI.js, 158
- DataView-more.js, 158
- FieldLabeler.js, 158
- FieldReplicator.js, 158
- Focus.js, 158
- GMapPanel.js, 158
- GroupSummary.js, 158
- GroupTab.js, 158
- GroupTabPanel.js, 158
- ItemSelector.js, 158
- LockingGridView.js, 158
- MultiSelect.js, 158
- PagingMemoryProxy.js, 158
- PanelResizer.js, 158
- Portal.js, 158
- PortalColumn.js, 158
- Portlet.js, 158
- ProgressBarPager.js, 158
- RowEditor.js, 158
- RowExpander.js, 158
- RowLayout.js, 158
- SearchField.js, 158
- SelectBox.js, 158
- SliderTip.js, 158
- SlidingPager.js, 158
- Spinner.js, 158
- SpinnerField.js, 158
- Spotlight.js, 158
- TabCloseMenu.js, 158
- TableGrid.js, 158
- TabScrollerMenu.js, 158
- ux-all-debug.js, 158
- ux-all.js, 158
- XmlTreeLoader.js, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/ux/data-view
 - CenterLayout.css, 158
 - ColumnHeaderGroup.css, 158
 - ColumnNodeUI.css, 158
 - GroupSummary.css, 158
 - GroupTab.css, 158
 - LockingGridView.css, 158
 - MultiSelect.css, 158
 - PanelResizer.css, 158
 - Portal.css, 158
 - RowEditor.css, 158
 - Spinner.css, 158
 - ux-all.css, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/ux/fileuploadfield
 - FileUploadField.js, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/ux/fileuploadfield/css
 - fileuploadfield.css, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/ux/gridfilters
 - GridFilters.js, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/ux/gridfilters/css
 - GridFilters.css, 158
 - RangeMenu.css, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/ux/gridfilters/filter
 - BooleanFilter.js, 158
 - DateFilter.js, 158
 - Filter.js, 158
 - ListFilter.js, 158
 - NumericFilter.js, 158
 - StringFilter.js, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/ux/gridfilters/menu
 - ListMenu.js, 158
 - RangeMenu.js, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/ux/statusbar
 - StatusBar.js, 158
 - ValidationStatus.js, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/ux/statusbar/css
 - statusbar.css, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/ux/treegrid
 - treegrid.css, 158
 - TreeGrid.js, 158
 - TreeGridColumnResizer.js, 158
 - TreeGridColumn.js, 158
 - TreeGridLoader.js, 158
 - TreeGridNodeUI.js, 158
 - TreeGridSorter.js, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/view
 - chooser-example.js, 158
 - chooser.css, 158
 - chooser.js, 158
 - data-view.css, 158
 - data-view.html, 158
 - data-view.js, 158
 - get-images.php, 158
 - list-view.css, 158
 - list-view.html, 158
 - list-view.js, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/window
 - gmap.js, 158
 - hello.js, 158
 - layout.js, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/writer
 - app.php, 158
 - UserForm.js, 158

- UserGrid.js, 158
- writer-thumb.gif, 158
- writer.css, 158
- writer.js, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/writer-tree.js, 158
- init.php, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/writer-tree.js, 158
- users.php, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/writer-tree.js, 158
- user.php, 158
- APIIS_HOME/lib/Apiis/Extjs/examples/writer-tree.js, 158
- application_controller.php, 158
- model.php, 158
- request.php, 158
- response.php, 158
- session_db.php, 158
- APIIS_HOME/lib/Apiis/Extjs/pkgs
- cmp-foundation-debug.js, 158
- cmp-foundation.js, 158
- data-foundation-debug.js, 158
- data-foundation.js, 158
- data-grouping-debug.js, 158
- data-grouping.js, 158
- data-json-debug.js, 158
- data-json.js, 158
- data-list-views-debug.js, 158
- data-list-views.js, 158
- data-xml-debug.js, 158
- data-xml.js, 158
- direct-debug.js, 158
- direct.js, 158
- ext-dd-debug.js, 158
- ext-dd.js, 158
- ext-foundation-debug.js, 158
- ext-foundation.js, 158
- pkg-buttons-debug.js, 158
- pkg-buttons.js, 158
- pkg-charts-debug.js, 158
- pkg-charts.js, 158
- pkg-forms-debug.js, 158
- pkg-forms.js, 158
- pkg-grid-editor-debug.js, 158
- pkg-grid-editor.js, 158
- pkg-grid-foundation-debug.js, 158
- pkg-grid-foundation.js, 158
- pkg-grid-grouping-debug.js, 158
- pkg-grid-grouping.js, 158
- pkg-grid-property-debug.js, 158
- pkg-grid-property.js, 158
- pkg-history-debug.js, 158
- pkg-history.js, 158
- pkg-menu-debug.js, 158
- pkg-menu.js, 158
- pkg-tabs-debug.js, 158
- pkg-tree-debug.js, 158
- pkg-tree.js, 158
- pkg-tips-debug.js, 158
- pkg-tips.js, 158
- pkg-toolbars-debug.js, 158
- pkg-toolbars.js, 158
- pkg-tree-debug.js, 158
- pkg-tree.js, 158
- resizable-debug.js, 158
- resizable.js, 158
- state-debug.js, 158
- state.js, 158
- window-debug.js, 158
- window.js, 158
- APIIS_HOME/lib/Apiis/Extjs/resources
- charts.swf, 158
- expressinstall.swf, 158
- resources.jsb, 158
- APIIS_HOME/lib/Apiis/Extjs/resources/css
- debug.css, 158
- ext-all-notheme.css, 158
- ext-all.css, 158
- README.txt, 158
- reset-min.css, 158
- xtheme-blue.css, 158
- xtheme-gray.css, 158
- APIIS_HOME/lib/Apiis/Extjs/resources/css/structure
- borders.css, 158
- box.css, 158
- button.css, 158
- combo.css, 158
- core.css, 158
- date-picker.css, 158
- dd.css, 158
- debug.css, 158
- dialog.css, 158
- editor.css, 158
- form.css, 158
- grid.css, 158
- layout.css, 158
- list-view.css, 158
- menu.css, 158
- panel-reset.css, 158
- panel.css, 158
- progress.css, 158
- qtips.css, 158
- reset.css, 158

- resizable.css, 158
- slider.css, 158
- tabs.css, 158
- toolbar.css, 158
- tree.css, 158
- window.css, 158
- APIIS_HOME/lib/Apiis/Extjs/resources/css/
 - borders.css, 158
 - box.css, 158
 - button.css, 158
 - combo.css, 158
 - core.css, 158
 - date-picker.css, 158
 - dd.css, 158
 - debug.css, 158
 - dialog.css, 158
 - editor.css, 158
 - form.css, 158
 - grid.css, 158
 - layout.css, 158
 - list-view.css, 158
 - menu.css, 158
 - panel.css, 158
 - progress.css, 158
 - qtips.css, 158
 - resizable.css, 158
 - slider.css, 158
 - tabs.css, 158
 - toolbar.css, 158
 - tree.css, 158
 - window.css, 158
- APIIS_HOME/lib/Apiis/Extjs/src
 - .DS_Store, 158
 - debug.js, 158
- APIIS_HOME/lib/Apiis/Extjs/src/adaptor
 - ext-base-dom-more.js, 158
 - jquery-bridge.js, 158
 - prototype-bridge.js, 158
 - yui-bridge.js, 158
- APIIS_HOME/lib/Apiis/Extjs/src/adaptor/Data
 - ext-base-ajax.js, 158
 - ext-base-anim-extra.js, 158
 - ext-base-anim.js, 158
 - ext-base-begin.js, 158
 - ext-base-dom.js, 158
 - ext-base-end.js, 158
 - ext-base-event.js, 158
 - ext-base-point.js, 158
 - ext-base-region.js, 158
- APIIS_HOME/lib/Apiis/Extjs/src/core
 - CompositeElement.js, 158
 - CompositeElementLite-more.js, 158
 - DomHelper-more.js, 158
 - Element-more.js, 158
 - Element.alignment.js, 158
 - Element.dd.js, 158
 - Element.fx-more.js, 158
 - Element.insertion-more.js, 158
 - Element.keys.js, 158
 - Element.legacy.js, 158
 - Element.position-more.js, 158
 - Element.scroll-more.js, 158
 - Element.style-more.js, 158
 - Element.traversal-more.js, 158
 - Error.js, 158
 - EventManager-more.js, 158
 - Ext-more.js, 158
 - Template-more.js, 158
- APIIS_HOME/lib/Apiis/Extjs/src/core/core
 - CompositeElementLite.js, 158
 - DomHelper.js, 158
 - DomQuery.js, 158
 - Element.fx.js, 158
 - Element.insertion.js, 158
 - Element.js, 158
 - Element.position.js, 158
 - Element.scroll.js, 158
 - Element.style.js, 158
 - Element.traversal.js, 158
 - EventManager.js, 158
 - Ext.js, 158
 - Fx.js, 158
 - Template.js, 158
- APIIS_HOME/lib/Apiis/Extjs/src/data
 - Api.js, 158
 - ArrayReader.js, 158
 - ArrayStore.js, 158
 - DataField.js, 158
 - DataProxy.js, 158
 - DataReader.js, 158
 - DataWriter.js, 158
 - DirectProxy.js, 158
 - DirectStore.js, 158
 - GroupingStore.js, 158
 - HttpProxy.js, 158
 - JsonReader.js, 158
 - JsonStore.js, 158
 - JsonWriter.js, 158
 - MemoryProxy.js, 158
 - Record.js, 158

- Request.js, 158
- Response.js, 158
- ScriptTagProxy.js, 158
- SortTypes.js, 158
- Store.js, 158
- StoreMgr.js, 158
- Tree.js, 158
- XmlReader.js, 158
- XmlStore.js, 158
- XmlWriter.js, 158
- APIIS_HOME/lib/Apiis/Extjs/src/data/core
 - Connection.js, 158
- APIIS_HOME/lib/Apiis/Extjs/src/dd
 - DDCore.js, 158
 - DragSource.js, 158
 - DragTracker.js, 158
 - DragZone.js, 158
 - DropTarget.js, 158
 - DropZone.js, 158
 - Registry.js, 158
 - ScrollManager.js, 158
 - StatusProxy.js, 158
- APIIS_HOME/lib/Apiis/Extjs/src/direct
 - Direct.js, 158
 - Event.js, 158
 - JsonProvider.js, 158
 - PollingProvider.js, 158
 - Provider.js, 158
 - RemotingProvider.js, 158
 - Transaction.js, 158
- APIIS_HOME/lib/Apiis/Extjs/src/localization
 - ext-lang-af.js, 158
 - ext-lang-bg.js, 158
 - ext-lang-ca.js, 158
 - ext-lang-cs.js, 158
 - ext-lang-da.js, 158
 - ext-lang-de.js, 158
 - ext-lang-el_GR.js, 158
 - ext-lang-en.js, 158
 - ext-lang-en_GB.js, 158
 - ext-lang-es.js, 158
 - ext-lang-fa.js, 158
 - ext-lang-fi.js, 158
 - ext-lang-fr.js, 158
 - ext-lang-fr_CA.js, 158
 - ext-lang-gr.js, 158
 - ext-lang-he.js, 158
 - ext-lang-hr.js, 158
 - ext-lang-hu.js, 158
 - ext-lang-id.js, 158
 - ext-lang-it.js, 158
 - ext-lang-ja.js, 158
 - ext-lang-ko.js, 158
 - ext-lang-lt.js, 158
 - ext-lang-lv.js, 158
 - ext-lang-mk.js, 158
 - ext-lang-nl.js, 158
 - ext-lang-no_NB.js, 158
 - ext-lang-no_NN.js, 158
 - ext-lang-pl.js, 158
 - ext-lang-pt.js, 158
 - ext-lang-pt_BR.js, 158
 - ext-lang-pt_PT.js, 158
 - ext-lang-ro.js, 158
 - ext-lang-ru.js, 158
 - ext-lang-sk.js, 158
 - ext-lang-sl.js, 158
 - ext-lang-sr.js, 158
 - ext-lang-sr_RS.js, 158
 - ext-lang-sv_SE.js, 158
 - ext-lang-th.js, 158
 - ext-lang-tr.js, 158
 - ext-lang-ukr.js, 158
 - ext-lang-vn.js, 158
 - ext-lang-zh_CN.js, 158
 - ext-lang-zh_TW.js, 158
- APIIS_HOME/lib/Apiis/Extjs/src/state
 - CookieProvider.js, 158
 - Provider.js, 158
 - StateManager.js, 158
- APIIS_HOME/lib/Apiis/Extjs/src/util
 - ClickRepeater.js, 158
 - Cookies.js, 158
 - CSS.js, 158
 - Date.js, 158
 - Format.js, 158
 - History.js, 158
 - KeyMap.js, 158
 - KeyNav.js, 158
 - MixedCollection.js, 158
 - Observable-more.js, 158
 - TextMetrics.js, 158
 - UpdateManager.js, 158
 - XTemplate.js, 158
- APIIS_HOME/lib/Apiis/Extjs/src/util/core
 - DelayedTask.js, 158
 - JSON.js, 158
 - Observable.js, 158
 - TaskMgr.js, 158
- APIIS_HOME/lib/Apiis/Extjs/src/widgets

- Action.js, 158
- BoxComponent.js, 158
- Button.js, 158
- ButtonGroup.js, 158
- ColorPalette.js, 158
- Component.js, 158
- ComponentMgr.js, 158
- Container.js, 158
- CycleButton.js, 158
- DataView.js, 158
- DatePicker.js, 158
- Editor.js, 158
- Layer.js, 158
- LoadMask.js, 158
- MessageBox.js, 158
- PagingToolbar.js, 158
- Panel.js, 158
- PanelDD.js, 158
- ProgressBar.js, 158
- Resizable.js, 158
- Shadow.js, 158
- Slider.js, 158
- SplitBar.js, 158
- SplitButton.js, 158
- TabPanel.js, 158
- Toolbar.js, 158
- Viewport.js, 158
- Window.js, 158
- WindowManager.js, 158
- APIIS_HOME/lib/Apiis/Extjs/src/widgets/
 - grid
 - AbstractSelectionModel.js, 158
 - CellSelectionModel.js, 158
 - CheckboxSelectionModel.js, 158
 - Column.js, 158
 - ColumnDD.js, 158
 - ColumnModel.js, 158
 - ColumnSplitDD.js, 158
 - EditorGrid.js, 158
 - GridDD.js, 158
 - GridEditor.js, 158
 - GridPanel.js, 158
 - GridView.js, 158
 - GroupingView.js, 158
 - PropertyGrid.js, 158
 - RowNumberer.js, 158
 - RowSelectionModel.js, 158
 - layout
 - AbsoluteLayout.js, 158
 - AccordionLayout.js, 158
 - AnchorLayout.js, 158
 - BorderLayout.js, 158
 - BoxLayout.js, 158
 - CardLayout.js, 158
 - ColumnLayout.js, 158
 - ContainerLayout.js, 158
 - FitLayout.js, 158
 - FormLayout.js, 158
 - FormTableLayout.js, 158
 - table
 - TableLayout.js, 158
- APIIS_HOME/lib/Apiis/Extjs/src/widgets/
 - chart
 - Chart.js, 158
 - event
 - EventProxy.js, 158
 - flash
 - FlashComponent.js, 158
 - swfobject.js, 158
- APIIS_HOME/lib/Apiis/Extjs/src/widgets/
 - action
 - Action.js, 158
 - basicform
 - BasicForm.js, 158
 - checkbox
 - Checkbox.js, 158
 - CheckboxGroup.js, 158
 - combo
 - Combo.js, 158
 - datefield
 - DateField.js, 158
 - displayfield
 - DisplayField.js, 158
 - field
 - Field.js, 158
 - fieldset
 - FieldSet.js, 158
 - form
 - Form.js, 158
 - hidden
 - Hidden.js, 158
 - htmleditor
 - HtmlEditor.js, 158
 - label
 - Label.js, 158
 - numberfield
 - NumberField.js, 158
 - radio
 - Radio.js, 158
 - RadioGroup.js, 158
 - textarea
 - TextArea.js, 158
 - textfield
 - TextField.js, 158
 - timefield
 - TimeField.js, 158
 - triggerfield
 - TriggerField.js, 158
 - vtypes
 - VTypes.js, 158
- APIIS_HOME/lib/Apiis/Extjs/src/widgets/list
 - Column.js, 158
 - ColumnResizer.js, 158
 - List View.js, 158
 - Sorter.js, 158
- APIIS_HOME/lib/Apiis/Extjs/src/widgets/menu
 - BaseItem.js, 158
 - CheckItem.js, 158
 - ColorMenu.js, 158
 - DateMenu.js, 158
 - Item.js, 158
 - Menu.js, 158
 - MenuMgr.js, 158
 - Separator.js, 158
 - TextItem.js, 158

- APIIS_HOME/lib/Apiis/Extjs/src/widgets/tip
 - QuickTip.js, 158
 - QuickTips.js, 158
 - Tip.js, 158
 - ToolTip.js, 158
- APIIS_HOME/lib/Apiis/Extjs/src/widgets/tree
 - AsyncTreeNode.js, 158
 - TreeDragZone.js, 158
 - TreeDropZone.js, 158
 - TreeEditor.js, 158
 - TreeEventModel.js, 158
 - TreeFilter.js, 158
 - TreeLoader.js, 158
 - TreeNode.js, 158
 - TreeNodeUI.js, 158
 - TreePanel.js, 158
 - TreeSelectionModel.js, 158
 - TreeSorter.js, 158
- APIIS_HOME/lib/Apiis/Form
 - erster.png, 158
 - erster2.png, 158
 - form.dtd, 158
 - Formdevelopment.tex, 158
 - HTML.pm, 158
 - CreateJSONData, 158
 - GetZIndex, 158
 - GetZIndexFile, 158
 - InitJSONData, 158
 - MakeStyle, 158
 - myfind, 158
 - PrintBody, 158
 - PrintForm, 158
 - PrintHeader, 158
 - PrintHeaderAjax, 158
 - PrintHeaderInit, 158
 - PrintMenu2, 158
 - PrintRowForm, 158
 - QueryJSONData, 158
 - ReduceStyle, 158
 - run, 158
 - RunAjax, 158
 - RunJSONEvents, 158
 - SaveJSONData, 158
 - Init.pm, 158
 - balloon_ref, 158
 - clear_block, 158
 - clear_form, 158
 - columnnames, 158
 - datasourcenames, 158
 - decode_list_ref, 158
 - DecValue, 158
 - fieldnames, 158
 - first_block, 158
 - font_string_for, 158
 - formnames, 158
 - generalnames, 158
 - get_bind_params, 158
 - get_block_ds, 158
 - get_event_par_ref, 158
 - get_field_data_ref, 158
 - get_field_list_ref, 158
 - insert_form, 158
 - last_block, 158
 - next_block, 158
 - prev_block, 158
 - PushEvent, 158
 - PushValue, 158
 - query_block, 158
 - runevents, 158
 - set_fielddtypes, 158
 - update_block, 158
 - update_blocks, 158
 - update_form, 158
 - new, 158
 - README.testing, 158
 - test.Form.ulf, 158
 - Tk.pm, 158
 - clear_form_error, 158
 - form_error, 158
 - run, 158
 - WWWUtilities.pm, 158
 - EvalCGIParameter, 158
 - SetDebug, 158
 - TestSystemForWWWAccess, 158
- APIIS_HOME/lib/Apiis/Form/Event
 - CallForm.pm, 158
 - callform, 158
 - Clear.pm, 158
 - HandledS.pm, 158
 - query_block, 158
 - Insert.pm, 158
 - Misc.pm, 158
 - printf_out.pm, 158
 - Query.pm, 158
 - create_msg, 158
 - Update.pm, 158
- APIIS_HOME/lib/Apiis/Form/forms
 - Ablammung.mfrm, 158
 - form.dtd, 158
 - navigationbar_fields.xml, 158

- q, 158
- RemoveBlanks.pl, 158
- test1.frm_, 158
- test9-scrapie.frm, 158
- APIIS_HOME/lib/Apiis/Form/HTML
 - CreateCSS.pm, 158
 - MakeStyle, 158
- Extjs.pm, 158
 - CreateArrayStore, 158
 - CreateDataArray, 158
 - CreateGridPanel, 158
 - PrintBody, 158
 - PrintHeaderInit, 158
- run, 158
- LabFrame.pm, 158
- Submit.pm, 158
- APIIS_HOME/lib/Apiis/Form/Init
 - Config.pm, 158
 - Misc.pm, 158
- APIIS_HOME/lib/Apiis/Form/Tk
 - BrowseEntry.pm, 158
 - Button.pm, 158
 - ButtonLib.pm, 158
 - do_last_block, 158
 - do_next_block, 158
 - do_prev_block, 158
 - do_runevents, 158
 - DateEntry.pm, 158
 - Label.pm, 158
 - LabFrame.pm, 158
 - Message.pm, 158
 - ScrollingList.pm, 158
 - Tabular.pm, 158
 - TextBlock.pm, 158
 - fetch_textblock, 158
 - store_textblock, 158
 - TextField.pm, 158
- APIIS_HOME/lib/Apiis/GUI
 - Ascii.pm, 158
 - MakeGUI, 158
 - PrintGUI, 158
 - PrintHeader, 158
- Excel.pm, 158
 - PrintAllCells, 158
 - PrintCell, 158
 - PrintGUI, 158
 - PrintHeader, 158
 - PrintObjects, 158
 - PrintRow, 158
 - PrintTable, 158
- FixPDF.pm, 158
 - colortrans, 158
 - MakeGUI, 158
 - PrintGUI, 158
 - PrintHeader, 158
- Hash.pm, 158
 - MakeGUI, 158
 - PrintGUI, 158
 - PrintHeader, 158
- HTML.pm, 158
 - Body, 158
 - MakeStyle, 158
 - new, 158
 - PrintBlock, 158
 - PrintCell, 158
 - PrintCellForm, 158
 - PrintHeader, 158
 - PrintObjects, 158
 - PrintRow, 158
 - PrintRowForm, 158
 - PrintTable, 158
- PDF.pm, 158
 - colortrans, 158
 - PrintCell, 158
 - PrintFooter, 158
 - PrintGUI, 158
 - PrintHeader, 158
 - PrintLongTable, 158
 - PrintObjects, 158
 - PrintRow, 158
 - PrintTable, 158
 - special_tex, 158
- XML.pm, 158
 - MakeGUI, 158
 - PrintGUI, 158
 - PrintHeader, 158
- APIIS_HOME/lib/Apiis/I18N
 - test.l10n, 152
- APIIS_HOME/lib/Apiis/Init
 - Cache.pm, 158
 - GetCache, 158
 - memcache, 158
 - new, 158
 - SetCache, 158
- Compat.pm, 158
 - get, 158
 - new, 158
 - set, 158
- Config.pm
 - start_tag, 158

- Date.pm, 158
- XML.pm, 158
 - AllStyleObjects, 158
 - basename, 158
 - CheckModul, 158
 - Children, 158
 - ContentFields, 158
 - CreateObjects, 158
 - Detail, 158
 - DetailObjects, 158
 - ext, 158
 - fullname, 158
 - General, 158
 - GetAllBlocks, 158
 - GetAllDataSources, 158
 - GetAllSubGUIs, 158
 - GroupFooter, 158
 - GroupFooterObjects, 158
 - GroupHeader, 158
 - GroupHeaderObjects, 158
 - GUI, 158
 - gui_file, 158
 - GUIFooter, 158
 - GUIFooterObjects, 158
 - GUIHeader, 158
 - GUIHeaderObjects, 158
 - MaxColumn, 158
 - MergeAttributes, 158
 - new, 158
 - PageFooter, 158
 - PageFooterObjects, 158
 - PageHeader, 158
 - PageHeaderObjects, 158
 - Parameter, 158
 - path, 158
 - SetColumnBusy, 158
 - XMLElements, 158
- XMLForms.pm.org, 158
 - allstyleobjects, 158
 - alt, 158
 - author, 158
 - autotab, 158
 - background, 158
 - backgroundattachment, 158
 - backgroundimage, 158
 - backgroundposition, 158
 - backgroundrepeat, 158
 - basename, 158
 - block, 158
 - blockfloat, 158
 - blockheight, 158
 - blocknames, 158
 - blockobjects, 158
 - blockwidth, 158
 - border, 158
 - borderbottom, 158
 - borderbottomwidth, 158
 - bordercolor, 158
 - borderleft, 158
 - borderleftwidth, 158
 - borderright, 158
 - borderrightwidth, 158
 - borderstyle, 158
 - bordertop, 158
 - bordertopwidth, 158
 - borderwidth, 158
 - button, 158
 - buttonlabel, 158
 - calendar, 158
 - CharSet, 158
 - charset, 158
 - checkbox, 158
 - checkboxgroup, 158
 - checked, 158
 - checkmodul, 158
 - children, 158
 - clear, 158
 - clip, 158
 - color, 158
 - column, 158
 - columnheader, 158
 - columnnames, 158
 - columnobjects, 158
 - columns, 158
 - command, 158
 - content, 158
 - currentblock, 158
 - cvs, 158
 - dataname, 158
 - dateref, 158
 - datasource, 158
 - datasourcenames, 158
 - datasourceobjects, 158
 - decimalplaces, 158
 - default, 158
 - description, 158
 - detailfield, 158
 - display, 158
 - dscolumn, 158
 - enabled, 158

event, 158
ext, 158
field, 158
fieldnames, 158
fieldobjects, 158
filefield, 158
floworder, 158
font, 158
fontfamily, 158
fontsize, 158
fontstyle, 158
fontunit, 158
fontvariant, 158
fontweight, 158
foreground, 158
format, 158
fullname, 158
function, 158
functionname, 158
general, 158
GetModul, 158
getobject, 158
GUI, 158
gui, 158
gui_file, 158
height, 158
image, 158
ishyperlink, 158
label, 158
labels, 158
left, 158
letterspacing, 158
line, 158
linebreak, 158
lineheight, 158
lineobjecttype, 158
linewidth, 158
link, 158
liststyle, 158
liststyleimage, 158
liststyleposition, 158
liststyletype, 158
margin, 158
marginbottom, 158
marginleft, 158
marginright, 158
margintop, 158
masterblock, 158
masterfield, 158
maxcolumn, 158
maxLength, 158
maxlength, 158
miscellaneous, 158
multiple, 158
name, 158
navigationbar, 158
objecttype, 158
onclick, 158
ondoubleclick, 158
ongetfocus, 158
onleavefocus, 158
order, 158
override, 158
padding, 158
paddingbottom, 158
paddingleft, 158
paddingright, 158
paddingtop, 158
parameter, 158
parent, 158
password, 158
path, 158
popupmenue, 158
position, 158
radiogroup, 158
record, 158
related, 158
row, 158
rowheader, 158
rows, 158
scrollinglist, 158
selected, 158
setcolumnbusy, 158
shiftcurrentblock, 158
size, 158
sql, 158
src, 158
statement, 158
StyleSheet, 158
stylesheet, 158
tabindex, 158
table, 158
tablename, 158
text, 158
textalign, 158
textarea, 158
textdecoration, 158
textfield, 158
textindent, 158
texttransform, 158

- top, 158
- type, 158
- unit, 158
- unshiftcurrentblock, 158
- url, 158
- version, 158
- verticalalign, 158
- visibility, 158
- whitespace, 158
- width, 158
- wordspacing, 158
- xmlelements, 158
- APIIS_HOME/lib/Apiis/Report
 - Base.pm, 150
 - GetData, 150
 - GetModelName, 150
 - LinkModul, 150
 - MakeReport, 150
 - new, 150
 - PrintFooter, 150
 - PrintReport, 150
 - Refresh, 150
 - ResetFooterObjects, 150
 - HTML.pm, 150
 - PrintCell, 150
 - PrintHeader, 150
 - PrintObjects, 150
 - PrintRow, 150
 - PrintTable, 150
 - Init.pm, 149
 - GetAllElementsXML, 149
 - GetAllParameter, 149
 - GetAllSubReports, 149
 - GetElementsXML, 149
 - Methods, 149
 - start, 149
 - InitXML.pm, 149
 - MergeAttributes, 149
 - new, 149
 - PDF.pm, 150
 - colortrans, 150
 - PrintCell, 150
 - PrintFooter, 150
 - PrintHeader, 150
 - PrintLongTable, 150
 - PrintObjects, 150
 - PrintReport, 150
 - PrintRow, 150
 - PrintTable, 150
- APIIS_HOME/lib/popreport
 - apiisrc, 158
 - dummy.dump, 158
- APIIS_HOME/lib/update
 - README.01, 158
 - upd_AR_01.txt, 158
 - upd_core_01.pl, 158
 - read_node, 158
 - xml_pretty_print, 158
 - upd_transfer_01.txt, 158
- Authorization
 - views, 11
- AUTHORS, 80, 98, 108
- check integrity
 - AUTHOR, 116
 - BUGS, 116
 - DESCRIPTION, 116
 - EXAMPLES, 116
 - OPTIONS, 116
 - SYNOPSIS, 115
 - VERSION, 116
- check ar, 80
- check integrity, 115
- check LastAction(), 98
- cl, 113
- CommaToDot, 93
 - DESCRIPTION, 93
 - SYNOPSIS, 93
- compare states, 108
- configuration, 113
- ConvertBool, 93
 - DESCRIPTION, 93
 - SYNOPSIS, 93
- LO keys, 131
- create a formatted form file from each
 - given loadobject by parsing
 - the loadobject file about vari-
 - able LO keys
 - AUTHOR, 131
 - EXAMPLE, 131
 - SEE ALSO, 131
 - SYNOPSIS, 131
- create client, 107
- create server, 108
- CreateCSSProperties, 112
- cu, 113
- cvs2cl pl
 - AUTHORS, 122
 - BUGS, 123
 - COPYRIGHT, 123

- DESCRIPTION, 117
- EXAMPLES, 121
- OPERATING SYSTEM COMPATIBILITY, 123
- OPTIONS, 117
- PREREQUISITES, 123
- SCRIPT CATEGORIES, 123
- SEE ALSO, 123
- SYNOPSIS, 117
- cvs2cl.pl, 117
- database structure, 9
- DataStream, 102
 - accessible (internal), 103
 - init (internal), 104
 - standard keys (internal), 103
- AUTHORS, 104
- CheckDS, 104
- data, 103
- debug, 103
- DESCRIPTION, 102
- ds, 102
- errors, 103
- ext unit, 103
- job end, 103
- job start, 102
- LO keys, 103
- new, 103
- PostHandling, 104
- PUBLIC METHODS, 102
- record seq, 103
- records error, 103
- records ok, 103
- records total, 103
- status, 103
- sth ds, 103
- sth inspool err, 103
- sth load stat, 103
- sth update inspool, 103
- SYNOPSIS, 102
- target column, 103
- verbose, 103
- DateDiff, 94
 - AUTHORS, 95
 - DESCRIPTION, 95
 - SYNOPSIS, 94
- DBCcreation pm
 - AUTHORS, 87
 - CreateDatabase, 87
 - DESCRIPTION, 87
 - InitDB, 87
 - LoadLang, 87
 - LoadNodeData, 87
 - SetSequences, 87
 - SUBROUTINES, 87
- DBCcreation.pm, 87
- DED state, 108
- DESCRIPTION, 69, 79
- description, 113, 114
- DotToColon, 93
 - DESCRIPTION, 94
 - SYNOPSIS, 93
- Fetch, 91
 - DESCRIPTION, 92
 - SYNOPSIS, 91
- file2inspool.pl, 124
- ForeignKey, 95
 - AUTHORS, 96
 - check ForeignKey(), 95
 - DESCRIPTION, 95
 - SYNOPSIS, 95
- Form, 124
 - AUTHOR, 125
 - DESCRIPTION, 124
 - OPTIONS, 125
 - SEE ALSO, 125
 - SYNOPSIS, 124
- format, 113
- FormDesigner, 125
 - AUTHOR, 130
 - checkSec, 129
 - clearGENERAL, 127
 - config EDLBNCATIOPMRU, 128
 - config unknown, 128
 - deleteField, 128
 - DESCRIPTION, 125
 - editsection, 128
 - fillColList, 128
 - loadForm, 126
 - loadmodel, 126
 - newForm, 126
 - opt fsel, 129
 - option apiis, 129
 - option debug, 129
 - option grid, 129
 - option sql, 129
 - options, 129
 - placeField, 128
 - Post, 129

- proceed, 128
 - renSection, 129
 - saveForm, 126
 - SEE ALSO, 130
 - selectFile, 126
 - selectFont, 127
 - SUBROUTINES, 126
 - SYNOPSIS, 125
 - textEdit, 128
 - updateField, 128
 - warnwin, 127
- initialize, 8
- ip2name, 108
- IsAFloat, 96
 - AUTHORS, 96
 - check IsAFloat(), 96
 - DESCRIPTION, 96
 - IsAFloat, 96
 - IsAFloat(), 96
 - SYNOPSIS, 96
- IsANumber, 96
 - is a number() (internal), 96
 - AUTHORS, 97
 - check IsANumber(), 96
 - DESCRIPTION, 96
 - IsANumber(), 96
 - SYNOPSIS, 96
- IsEqual, 97
 - AUTHORS, 97
 - check IsEqual(), 97
 - DESCRIPTION, 97
 - SYNOPSIS, 97
 - VERSION, 97
- LastAction, 97
- List, 98
 - AUTHORS, 99
 - check List(), 99
 - DESCRIPTION, 98
 - SYNOPSIS, 98
- load merge element, 108
- LowerCase, 94
 - AUTHORS, 94
 - DESCRIPTION It is usually used as a MODIFY-rule in the model file., 94
 - SYNOPSIS, 94
- METHODS, 70
- mkform, 130
 - AUTHOR, 130
 - DESCRIPTION, 130
 - FILES, 130
 - OPTIONS, 130
 - SYNOPSIS, 130
- mkLOform, 131
 - AUTHOR, 132
 - EXAMPLE, 131
 - SEE ALSO, 132
 - SYNOPSIS, 131
- mksql – create SQL commands from the model file, 132
 - AUTHOR, 132
 - COPYRIGHT, 132
 - DESCRIPTION, 132
 - OPTIONS, 132
 - SEE ALSO, 132
 - SYNOPSIS, 132
- Model file, 9
- model2xfig, 132
 - AUTHOR, 133
 - DESCRIPTION, 133
 - OPTIONS, 133
 - SEE ALSO, 133
 - SYNOPSIS, 132
- n, 113
- name2ip, 108
- new, 107
- NoCheck, 99
 - AUTHORS, 99
 - check NoCheck(), 99
 - DESCRIPTION, 99
 - NoCheck(), 99
 - SYNOPSIS, 99
- NoNumber, 99
 - AUTHORS, 99
 - check NoNumber(), 99
 - DESCRIPTION, 99
 - NoNumber(), 99
 - SYNOPSIS, 99
- NotNull, 100
 - AUTHORS, 100
 - DESCRIPTION, 100
 - SYNOPSIS, 100
- popup, 112
- position, 113
- PseudoSQL, 15

- PseudoStatement, 105
 - accessible (internal), 105
 - init (internal), 106
 - standard keys (internal), 105
 - actionname, 105
 - AUTHORS, 106
 - column extfields, 105
 - columns, 105
 - DESCRIPTION, 105
 - errors, 105
 - extfields, 105
 - new, 105
 - PUBLIC METHODS, 105
 - pull quotes, 106
 - status, 105
 - SYNOPSIS, 105
 - tablename, 105
 - value, 105
 - values, 105
 - whereclause, 105
- PUBLIC METHODS, 107
- Range, 100
 - AUTHORS, 100
 - BUGS, 100
 - check Range(), 100
 - DESCRIPTION, 100
 - Range(), 100
 - SYNOPSIS, 100
- read DED, 108
- redo clases, 80
- ReservedStrings, 100
 - AUTHORS, 101
 - check ReservedStrings, 100
 - DESCRIPTION, 100
 - SYNOPSIS, 100
- return, 114
- SELECT, 15
- SetGuid, 101
 - AUTHORS, 101
 - check SetGuid(), 101
 - SetGuid(), 101
 - SYNOPSIS, 101
- SetNode, 102
 - AUTHORS, 102
 - check SetNode(), 102
 - SetNode(), 102
 - SYNOPSIS, 102
- SetNow, 94
 - DESCRIPTION, 94
 - SYNOPSIS, 94
- SetUser, 94
 - DESCRIPTION, 94
 - SYNOPSIS, 94
- SetVersion, 102
 - AUTHORS, 102
 - check SetVersion(), 102
 - SetVersion(), 102
 - SYNOPSIS, 102
- show rules
 - h, 133
 - o <outfile>, 134
 - p <project>, 134
 - v, 134
 - description, 133
 - options, 133
 - usage, 133
- show rules, 133
- skip LastAction(), 98
- Statement, 106
 - accessible (internal), 107
 - init (internal), 107
 - standard keys (internal), 106
 - actionname, 106
 - AUTHORS, 107
 - columns, 106
 - DESCRIPTION, 106
 - errors, 106
 - new, 107
 - PUBLIC METHODS, 106
 - status, 106
 - SYNOPSIS, 106
 - tablename, 106
 - value, 106
 - values, 106
 - whereclause, 106
- SUBROUTINES, 79
- SYNOPSIS, 69, 107
- Unique, 101
 - AUTHORS, 101
 - check Unique(), 101
 - DESCRIPTION, 101
 - SYNOPSIS, 101
 - Unique, 101
 - Unique(), 101
- UpperCase, 94
 - AUTHORS, 94

- DESCRIPTION It is usually used
as a MODIFY-rule in the
model file., 94
- SYNOPSIS, 94
- usage, 113, 114
- used in, 113
- WebForm.pl
 - ABSTRACT, 134
 - AUTHOR INFORMATION, 135
 - BUGS, 135
 - Comment, 135
 - Configuration, 134
 - Notice, 134
 - PROGRAMMING STYLE, 134
 - SEE ALSO, 135
- WebForm.pl, 134
- xfig lib pm
 - arrowlinetype, 115
 - AUTHOR, 115
 - boxtype, 115
 - compoundtype, 115
 - DESCRIPTION, 114
 - getFileHeader, 114
 - linetype, 115
 - SEE ALSO, 115
 - SUBROUTINES, 114
 - texttype, 114
- xfig lib.pm, 114