Programme Area: Carbon Capture and Storage

Project: Storage Appraisal

Title: Technical Specification for the UKSAP WDG Application

Abstract:

This document is a supporting document to deliverable MS6.1 UK Storage Appraisal Final Report.

Context:

This £4m project produced the UK's first carbon dioxide storage appraisal database enabling more informed decisions on the economics of CO2 storage opportunities. It was delivered by a consortium of partners from across academia and industry - LR Senergy Limited, BGS, the Scottish Centre for Carbon Storage (University of Edinburgh, Heriot-Watt University), Durham University, GeoPressure Technology Ltd, Geospatial Research Ltd, Imperial College London, RPS Energy and Element Energy Ltd. The outputs were licensed to The Crown Estate and the British Geological Survey (BGS) who have hosted and further developed an online database of mapped UK offshore carbon dioxide storage capacity. This is publically available under the name CO2 Stored. It can be accessed via www.co2stored.co.uk.

Disclaimer:

The Energy Technologies Institute is making this document available to use under the Energy Technologies Institute Open Licence for Materials. Please refer to the Energy Technologies Institute website for the terms and conditions of this licence. The Information is licensed 'as is' and the Energy Technologies Institute excludes all representations, warranties, obligations and liabilities in relation to the Information to the maximum extent permitted by law. The Energy Technologies Institute is not liable for any errors or omissions in the Information and shall not be liable for any loss, injury or damage of any kind caused by its use. This exclusion of liability includes, but is not limited to, any direct, indirect, special, incidental, consequential, punitive, or exemplary damages in each case such as loss of revenue, data, anticipated profits, and lost business. The Energy Technologies Institute does not guarantee the continued supply of the Information. Notwithstanding any statement to the contrary contained on the face of this document, the Energy Technologies Institute confirms that the authors of the document have consented to its publication by the Energy Technologies Institute.

UKSAP

Appendix A8.1

Technical Specification for the UKSAP WDG Application

Conducted for

The Energy Technologies Institute

By

Senergy Survey and GeoEngineering

Authors	Richard Orren
Technical Audit	
Quality Audit	
Release to Client	Grahame Smith
Date Released	28th October 2011 (final)

The Consortium has made every effort to ensure that the interpretations, conclusions and recommendations presented herein are accurate and reliable in accordance with good industry practice and its own quality management procedures. The Consortium does not, however, guarantee the correctness of any such interpretations and shall not be liable or responsible for any loss, costs, damages or expenses incurred or sustained by anyone resulting from any interpretation or recommendation made by any of its officers, agents or employees.

File Location
Appendix A8.1 - Technical Specification for the UKSAP WDG Application.doc

Contents

1. Technical Specification for the UKSAP WDG Application

1.1 Map Server

Esri ArcGIS Server 10 is required to be installed, this can be on a different server from the application server. Currently the Workgroup version of ArcGIS Java Server 10 is being used, though it should also be possible to use the .Net version of the server as well.

The javascript API must be enabled from the REST web control panel on the ArcGIS server.

The measure tool must be enabled and requires the REST service to be active.

If Esri ArcGIS server 9 is used, a competent JavaScript programmer would need to change the ESRI JavaScript API used to an earlier version, otherwise the measure tool will not work.

1.2 Database

This is an SQL Server Database. The free SQL Server Express has all the features needed for this Database.

Table Name	Number of Rows (as of Aug 2011)	Expected to Grow?
allunitsdata	Insignificant	No
author	50000	Yes. Every time a page of storage unit data is edited, a record is created.
bugs	Insignificant	Minor. Every time a bug is submitted through the form on the website
capacity	8000	Yes. Every time a page of storage unit data with records on this table is edited, a record is created.
comment	5000	Minor. There is 1 row for every comment.
computedvalues	600	Minor. There is 1 row for every storage unit.
databaseupgrades	Insignificant	No
economics	10	Yes. Every time a page of storage unit data with records on this table is edited, a record is created.
featurerequests	Insignificant	Minor. Every time a bug is submitted through the form on the website
general	5000	Yes. Every time a page of storage unit data with records on this table is edited, a record is created.
htmlpages	Insignificant	No
montecarlodistribution	2000000	Minor. There is a set number of rows for every storage unit, and the number of storage units is not expected to grow.
montecarlomean	90000	"
montecarlomedian	90000	"

Table Name	Number of Rows (as of Aug 2011)	Expected to Grow?
montecarlomode	90000	"
montecarlopvalues	800000	"
montecarlostrandarddev	90000	"
montecarlovariance	90000	"
oilgaswell	3000	Yes. Every time a page of storage unit data with records on this table is edited, a record is created.
porevolume	5500	Yes. Every time a page of storage unit data with records on this table is edited, a record is created.
risk	10000	Yes. Every time a page of storage unit data with records on this table is edited, a record is created.
source	200	Every time a source is submitted through the form on the website
staticpages	Insignificant	No
storageunit	660	Minor. There is 1 row for every storage unit.
user	80	Minor. There is 1 row for every user.

1.3 Application Server

1.3.1.1 Language

PHP5 is used.

1.4 Web Server

CarbonStore is tested with IIS7, but in theory it should be possible to use Apache.

We would recommend client-side caching of CSS, JavaScript and all image files configured in the web server.

1.5 ODBC

An ODBC connection is used to connect to the Database. This must be the same as the PHP installation; if 32 bit PHP is being used a 32-bit ODBC connection must be used .

The ODBC name must be the same as the Database name.

This means that if it is required to have multiple Carbonstore installs (e.g to retain one for testing) a separate ODBC connection will be needed per install. (as will a separate database, file storage area and background processes.)

1.6 File Storage

An area of disk space must be readable by the web-server. It must not be served by the web-server, otherwise security measures could be bypassed. Currently, Carbonstore uses 20MB of disk space and as this is directly related to the number of storage units it is not expected to grow significantly.

1.7 Background Processes- Run Regularly

There are several background processes that must be run regularly. They are used to build CSVs or perform calculations. If the site is being edited regularly they must be run regularly. If the data on the site is not being edited, they must be run until all calculations are done then they can be turned off.

They are all in PHP and need access to the Database through an ODBC connection.

Some of them need write access to the file storage area mentioned above.

Script	Purpose	Needs File Storage Access	Run	Load
allUnits.php	Generates some statistics about all units.	No	Regularly	Insignificant
cacheProgressControl.php	Calculates Progress Control data for each unit and caches it.	No	Regularly	Minor
emailPendingReferenceSources.php	Emails pending reference sources to site administrators. Will email every time it is run (if there are any sources to send) so only run once per day.	No	Once per day	Insignificant
GenerateAllStorageUnitsAsCSV.php	Builds CSV files of all storage units and stores in File Storage area for download later.	Yes	Regularly	Insignificant

As noted, emailPendingReferenceSources.php should only be run once per day (possibly only on week days). For the other scripts, it is a balance between how long you want to leave the application with out-dated or no data and the load on the server.

Note background processes can be run on a different server from the webserver, as long as they have access to the same database and file storage.

1.8 Background Process – Run Continuously

The calculator process calculates mathematical results for each storage unit, sometimes using Monte Carlo analysis.

It must be run continuously. It is currently configured to run in a command prompt window.

Ideally, a version would be available to run as a Windows Service to.

Note this can be run on a different server from the webserver, as long as it has access to the same database.

1.9 Emails

The program seeks to send emails, both from the actual webserver and from the EmailPendingReferenceSources.php background process script, using the normal PHP mail() function. Access to an email server is needed.

1.10 Backup

The database needs to be backed up regularly. File storage areas on the server do not as they can be rebuilt quickly.

2. PHP Class Containing All Formula, Algorithms and Constants

Class StorageUnitCalculator
This module contains nothing but static methods to do mathematical calculations. Parameters are input variables, results are returned. There is no interaction with Data Objects or the Database; so it can be used from normal calculations and Monte Carlo probabilistic calculations.

Located in /includes/StorageUnitCalculator.class.php (line 7)

Class Constant Summary
BRINE_DENSITY = 1.026
ESTIMATED_INJECTION_RATE_PER_WELL_MAX = 3.5
ESTIMATED_INJECTION_RATE_PER_WELL_MIN = 0.1
GRAVITY = 9.8067
MAX_ALLOWABLE_STORAGE_PRESSURE_AS_A_FRACTION_OF_FRACTURE_PRESSURE = 0.9
MAX_CUMULATIVE_INJECTION_RATE_PER_WELL_MAX = 20
MAX_CUMULATIVE_INJECTION_RATE_PER_WELL_MIN = 0.75
MIGRATION_VELOCITY_THRESHHOLD = 10
PERMEABILITY_THRESHHOLD = 10
PVMULTCOMBINEDDEV_MAX_ADDITION = 33
PVMULTCOMBINEDDEV_MAX_MULT = 0.0
PVMULTCOMBINEDDEV_MIN_ADDITION = 1
PVMULTCOMBINEDDEV_MIN_MULT = 0.0
PVMULTCOMBINEDDEV_ML_ADDITION = 20.3
PVMULTCOMBINEDDEV_ML_MULT = -0.897
REPRESENTATIVE_FRACTURE_PRESSURE_GRADIENT = 19
REPRESENTATIVE_LITHOSTATIC_PRESSURE_GRADIENT = 20
SALINE_WATER_PRESSURE_GRADIENT = 10.0659
SURFACE_TEMPERATURE = 8

Variable Summary
static mixed $cacheLookUpTables

Method Summary
static void calcAquiferCapacity ($lithostaticpressure, $fracturepressure, $porepressure)
static void calcAveragePorePressureAtCentroid ($finalporepressureatcentroid, $initialporepressureatcentroid)
static void calcBuoyantTrappingCapacity ($poreVolume, $sweepEfficiency, $irreducibleWaterSaturation, $codensity, $isParentUnitFullyConfined, $parentUnitPressureCapacity, $grossthickness, $estimatedRelief, $co2height)
static void calcCO2DensityAtRefDepthAndVirginReservoirConditions ($virginReservoirPressureAtReferenceDepth, $virginReservoirTemperatureAtReferenceDepth)
static void calcCO2PressureGradientAtStorage ($codensity)
static void calcCO2ViscosityAtRefDepthAndVirginReservoirConditions ($virginReservoirPressureAtReferenceDepth, $virginReservoirTemperatureAtReferenceDepth)
static void calcCODensity ($finalporepressureatcentroid, $formationtempatcentrioddepth)
static void calcCOHeight ($aquifercapacity, $co2pressuregradientatstorage)
static void calcCombinedDevelopmentCapacity ($poreVolume, $PVMultCombinedDev, $codensity)
static void calcCOViscosity ($averageporepressureatcentroid, $formationtempatcentrioddepth)
static void calcCumulativeGasProduction ($cumulativeSalesGas, $cumulativeFuelFlareGas)
static void calcEstimatedInjectionRatePerWellForGas ($AverageHydrocarbonProductionRatePerWell, $GasFormationVolumeFactor,

```
$CO2DensityAtReferenceDepth, $GasViscosityAtReservoirConditions,
$CO2ViscosityAtReferenceDepth, $waterFormationVolumeFactor,
$AverageWaterProductionRatePerWell)
static void calcEstimatedInjectionRatePerWellForGasCondensate (
$AverageHydrocarbonProductionRatePerWell, $GasShrinkageFactor,
$GasFormationVolumeFactor, $CO2DensityAtReferenceDepth,
$GasViscosityAtReservoirConditions, $CO2ViscosityAtReferenceDepth,
$waterFormationVolumeFactor, $AverageWaterProductionRatePerWell)
static void calcEstimatedInjectionRatePerWellForOilAndGas (
$AverageHydrocarbonProductionRatePerWell, $OilFormationVolumeFactor,
$CO2DensityAtReferenceDepth, $OilViscosityAtReservoirConditions,
$CO2ViscosityAtReferenceDepth, $waterFormationVolumeFactor,
$AverageWaterProductionRatePerWell)
static void calcFinalMaxPorePressureShallowest ( $lithostaticpressure, $fracturepressure,
$isFullyConfinedClosedBoxStorageUnit, $isStructuralStratigraphicTrapStorageUnit,
$porepressure)
static void calcFinalPorePressureAtCentroid ( $finalmaxporepressureshallowest,
$meandepth, $shallowestdepth)
static void calcFluidViscocity ( $averageporepressureatcentroid,
$formationtempatcentrioddepth, $salinity)
static void calcFormationTempAtCentriodDepth ( $temperaturegradient, $meandepth)
static void calcFormationWaterCompress ( $finalporepressureatcentroid,
$formationtempatcentrioddepth, $salinity)
static void calcFracturePresureAtCentroidDepth ( $fracturePressureAtShallowestdepth,
$centroidDepth, $shallowestDepth)
static void calcGasCondensateStorageCapacity ( $cumulativeGasProduction,
$gasShrinkageFactor, $gasFormationVolumeFactor, $cumulativeCondensateProduction,
$condensateGasRatioAtVirginReservoirConditions, $condensateFormationVolumeFactor,
$cumulativeWaterProduction, $waterFormationVolumeFactor, $cumulativeGasInjection,
$CO2DensityAtRefDepthAndVirginReservoirConditions)
static void calcGasFormationVolumeFactor ( $gasCompressibilityFactorAtPAndT,
$virginReservoirPressureAtReferenceDepth,
$virginReservoirTemperatureAtReferenceDepth)
static void calcGasShrinkageFactor ( $APIofGasCondensateAtVirginReservoirConditions,
$condensateGasRatioAtVirginReservoirConditions)
static void calcGasStorageCapacity ( $cumulativeGasProduction,
$gasFormationVolumeFactor, $cumulativeWaterProduction, $waterFormationVolumeFactor,
$cumulativeGasInjection, $CO2DensityAtRefDepthAndVirginReservoirConditions)
static void calcGrv ( $area, $grossthickness, $estimatedRelief, $shapeFactor,
$isStructuralStratigraphicTrap, $isSalineAquifer)
static void calcHydrostatic ( $shallowestdepth)
static void calcInitialPorePressureAtCentroid ( $porepressure, $meandepth,
$shallowestdepth)
static void calcLithostaticPresureAtCentroidDepth ( $lithostaticPressureAtShallowestdepth,
$centroidDepth, $shallowestDepth)
static void calcMaxCumulativeInjectionPerWellForGas ( $MaxCumulativeProductionPerWell,
$GasFormationVolumeFactor, $CO2DensityAtReferenceDepth)
static void calcMaxCumulativeInjectionPerWellForGasCondensate (
$MaxCumulativeProductionPerWell, $GasShrinkageFactor, $GasFormationVolumeFactor,
$CO2DensityAtReferenceDepth)
static void calcMaxCumulativeInjectionPerWellForOilAndGas (
$MaxCumulativeProductionPerWell, $OilFormationVolumeFactor,
$CO2DensityAtReferenceDepth)
static void calcMaxPressure ( $lithostaticPressureAtCentroidDepth,
$fracturePressureAtCentroidDepth)
static void calcMinMlMaxForPVMultCombinedDev ( $numberOfSalineDaughters)
static void calcNumberWellsForHydroCarbonInjectivity ( $inj, $dur,
$hydrocarbonStorageCapacity, $estimatedInjRatePerWell, $maxCumInjPerWell)
static void calcOilGasStorageCapacity ( $cumulativeOilProduction,
$oilFormationVolumeFactor, $cumulativeGasProduction,
$solutionGasOilRatioAtVirginReservoirConditions, $gasFormationVolumeFactor,
```

$cumulativeWaterProduction, $waterFormationVolumeFactor, $cumulativeWaterInjection, $cumulativeGasInjection, $CO2DensityAtRefDepthAndVirginReservoirConditions)
static void calcOpenAquiferParentCapacity ($poreVolume, $poreSpaceUtilisation, $pc02)
static void calcPoreSpaceUtilisationForTicket124 ($PVMultCombinedDev, $swirr, $volumeSweep)
static void calcPoreVolume ($grv, $arealsand, $ntg, $porosity)
static void calcPredictedMigrationVelocity ($storagePermeability, $pc02, $dip, $krco2, $uco2, $averagePorosity, $swirr)
static void calcPressureCapacity ($porevolume, $formationwatercompress, $rockcompress, $aquifercapacity, $codensity)
static void calcPressureForHydroCarbonInjectivity ($referenceDepth)
static void calcPVMultFullyConfined ($maxPressure, $initalPorePressure, $totalCompressibility)
static void calcPVxEfactor ($storageefficiency, $codensity, $porevolume)
static void calcRockCompressMax ($averagePorosityML)
static void calcRockCompressMin ($averagePorosityML)
static void calcRockCompressML ($averagePorosityML)
static void calcScaledFormationArea ($formationArea, $PoreSpaceUtilisation, $PVMult_FullyConfined)
static void calcTemperatureGradient ($formationtemp, $shallowestdepth)
static void calcThicknessArea ($grossthickness, $area)
static void calcTotalCompressibility ($rockCompressibility, $formationWaterCompressibility)
static void cleanResult ($in)
static void find2NearestPointsInArray ($array, $desired)
static void getLookUpTable ($name)
static void getValueFromLookUpTable ($table, $X, $Y)
static void interpolateBetween2Values ($value1idx, $value1data, $value2idx, $value2data, $desired)

Variables
static mixed $cacheLookUpTables = array() (line 83)

access: public

Methods
static calcAquiferCapacity (line 177)

access: public

void calcAquiferCapacity ($lithostaticpressure, $fracturepressure, $porepressure)

$lithostaticpressure
$fracturepressure
$porepressure

static calcAveragePorePressureAtCentroid (line 170)

access: public

void calcAveragePorePressureAtCentroid ($finalporepressureatcentroid, $initialporepressureatcentroid)

$finalporepressureatcentroid
$initialporepressureatcentroid

static calcBuoyantTrappingCapacity (line 448)

access: public

void calcBuoyantTrappingCapacity ($poreVolume, $sweepEfficiency, $irreducibleWaterSaturation, $codensity, $isParentUnitFullyConfined, $parentUnitPressureCapacity, $grossthickness, $estimatedRelief, $co2height)

$poreVolume
$sweepEfficiency
$irreducibleWaterSaturation
$codensity
$isParentUnitFullyConfined
$parentUnitPressureCapacity
$grossthickness
$estimatedRelief
$co2height

static calcCO2DensityAtRefDepthAndVirginReservoirConditions (line 359)

access: public

void calcCO2DensityAtRefDepthAndVirginReservoirConditions ($virginReservoirPressureAtReferenceDepth, $virginReservoirTemperatureAtReferenceDepth)

$virginReservoirPressureAtReferenceDepth
$virginReservoirTemperatureAtReferenceDepth

static calcCO2PressureGradientAtStorage (line 213)

access: public

void calcCO2PressureGradientAtStorage ($codensity)

$codensity

static calcCO2ViscosityAtRefDepthAndVirginReservoirConditions (line 374)

defined ticket 134.

defined ticket 134. *

access: public

void calcCO2ViscosityAtRefDepthAndVirginReservoirConditions ($virginReservoirPressureAtReferenceDepth, $virginReservoirTemperatureAtReferenceDepth)

$virginReservoirPressureAtReferenceDepth
$virginReservoirTemperatureAtReferenceDepth

static calcCODensity (line 199)

access: public

void calcCODensity ($finalporepressureatcentroid, $formationtempatcentrioddepth)

$finalporepressureatcentroid
$formationtempatcentrioddepth

static calcCOHeight (line 235)

access: public

void calcCOHeight ($aquifercapacity, $co2pressuregradientatstorage)

$aquifercapacity
$co2pressuregradientatstorage

static calcCombinedDevelopmentCapacity (line 465)

access: public

void calcCombinedDevelopmentCapacity ($poreVolume, $PVMultCombinedDev, $codensity)

$poreVolume
$PVMultCombinedDev
$codensity

static calcCOViscosity (line 220)

access: public

void calcCOViscosity ($averageporepressureatcentroid, $formationtempatcentrioddepth)

$averageporepressureatcentroid
$formationtempatcentrioddepth

static calcCumulativeGasProduction (line 332)

Ticket 91, if only one variable known we calculate anyway.

Ticket 91, if only one variable known we calculate anyway. *

access: public

void calcCumulativeGasProduction ($cumulativeSalesGas, $cumulativeFuelFlareGas)

$cumulativeSalesGas
$cumulativeFuelFlareGas

static calcEstimatedInjectionRatePerWellForGas (line 572)

defined ticket 134.

defined ticket 134. *

access: public

void calcEstimatedInjectionRatePerWellForGas ($AverageHydrocarbonProductionRatePerWell, $GasFormationVolumeFactor, $CO2DensityAtReferenceDepth, $GasViscosityAtReservoirConditions, $CO2ViscosityAtReferenceDepth, $waterFormationVolumeFactor, $AverageWaterProductionRatePerWell)

$AverageHydrocarbonProductionRatePerWell
$GasFormationVolumeFactor
$CO2DensityAtReferenceDepth
$GasViscosityAtReservoirConditions
$CO2ViscosityAtReferenceDepth
$waterFormationVolumeFactor
$AverageWaterProductionRatePerWell

static calcEstimatedInjectionRatePerWellForGasCondensate (line 595) defined ticket 134.

defined ticket 134. *

access: public

void calcEstimatedInjectionRatePerWellForGasCondensate ($AverageHydrocarbonProductionRatePerWell, $GasShrinkageFactor, $GasFormationVolumeFactor, $CO2DensityAtReferenceDepth, $GasViscosityAtReservoirConditions, $CO2ViscosityAtReferenceDepth, $waterFormationVolumeFactor, $AverageWaterProductionRatePerWell)

$AverageHydrocarbonProductionRatePerWell
$GasShrinkageFactor
$GasFormationVolumeFactor
$CO2DensityAtReferenceDepth
$GasViscosityAtReservoirConditions
$CO2ViscosityAtReferenceDepth
$waterFormationVolumeFactor
$AverageWaterProductionRatePerWell

static calcEstimatedInjectionRatePerWellForOilAndGas (line 549)

defined ticket 134.

defined ticket 134. *

access: public

void calcEstimatedInjectionRatePerWellForOilAndGas ($AverageHydrocarbonProductionRatePerWell, $OilFormationVolumeFactor, $CO2DensityAtReferenceDepth, $OilViscosityAtReservoirConditions, $CO2ViscosityAtReferenceDepth, $waterFormationVolumeFactor, $AverageWaterProductionRatePerWell)

$AverageHydrocarbonProductionRatePerWell
$OilFormationVolumeFactor
$CO2DensityAtReferenceDepth
$OilViscosityAtReservoirConditions
$CO2ViscosityAtReferenceDepth
$waterFormationVolumeFactor
$AverageWaterProductionRatePerWell

static calcFinalMaxPorePressureShallowest (line 152)

access: public

void calcFinalMaxPorePressureShallowest ($lithostaticpressure, $fracturepressure, $isFullyConfinedClosedBoxStorageUnit, $isStructuralStratigraphicTrapStorageUnit, $porepressure)

$lithostaticpressure
$fracturepressure
$isFullyConfinedClosedBoxStorageUnit
$isStructuralStratigraphicTrapStorageUnit
$porepressure

static calcFinalPorePressureAtCentroid (line 163)

access: public

void calcFinalPorePressureAtCentroid ($finalmaxporepressureshallowest, $meandepth, $shallowestdepth)

   $finalmaxporepressureshallowest
   $meandepth
   $shallowestdepth

static calcFluidViscocity (line 285)

   access: public

void calcFluidViscocity ($averageporepressureatcentroid, $formationtempatcentrioddepth, $salinity)

   $averageporepressureatcentroid
   $formationtempatcentrioddepth
   $salinity

static calcFormationTempAtCentriodDepth (line 192)

TODO: This function is spelled wrong *

TODO: This function is spelled wrong *

   access: public

void calcFormationTempAtCentriodDepth ($temperaturegradient, $meandepth)

   $temperaturegradient
   $meandepth

static calcFormationWaterCompress (line 242)

   access: public

void calcFormationWaterCompress ($finalporepressureatcentroid, $formationtempatcentrioddepth, $salinity)

   $finalporepressureatcentroid
   $formationtempatcentrioddepth
   $salinity

static calcFracturePresureAtCentroidDepth (line 506)

defined ticket 124 *

defined ticket 124 *

   access: public

void calcFracturePresureAtCentroidDepth ($fracturePressureAtShallowestdepth, $centroidDepth, $shallowestDepth)

   $fracturePressureAtShallowestdepth
   $centroidDepth
   $shallowestDepth

static calcGasCondensateStorageCapacity (line 428)

   access: public

void calcGasCondensateStorageCapacity ($cumulativeGasProduction, $gasShrinkageFactor, $gasFormationVolumeFactor, $cumulativeCondensateProduction, $condensateGasRatioAtVirginReservoirConditions, $condensateFormationVolumeFactor, $cumulativeWaterProduction, $waterFormationVolumeFactor, $cumulativeGasInjection, $CO2DensityAtRefDepthAndVirginReservoirConditions)

$cumulativeGasProduction
$gasShrinkageFactor
$gasFormationVolumeFactor
$cumulativeCondensateProduction
$condensateGasRatioAtVirginReservoirConditions
$condensateFormationVolumeFactor
$cumulativeWaterProduction
$waterFormationVolumeFactor
$cumulativeGasInjection
$CO2DensityAtRefDepthAndVirginReservoirConditions

static calcGasFormationVolumeFactor (line 343)

access: public

void calcGasFormationVolumeFactor ($gasCompressibilityFactorAtPAndT, $virginReservoirPressureAtReferenceDepth, $virginReservoirTemperatureAtReferenceDepth)

$gasCompressibilityFactorAtPAndT
$virginReservoirPressureAtReferenceDepth
$virginReservoirTemperatureAtReferenceDepth

static calcGasShrinkageFactor (line 351)

access: public

void calcGasShrinkageFactor ($APIofGasCondensateAtVirginReservoirConditions, $condensateGasRatioAtVirginReservoirConditions)

$APIofGasCondensateAtVirginReservoirConditions
$condensateGasRatioAtVirginReservoirConditions

static calcGasStorageCapacity (line 411)

access: public

void calcGasStorageCapacity ($cumulativeGasProduction, $gasFormationVolumeFactor, $cumulativeWaterProduction, $waterFormationVolumeFactor, $cumulativeGasInjection, $CO2DensityAtRefDepthAndVirginReservoirConditions)

$cumulativeGasProduction
$gasFormationVolumeFactor
$cumulativeWaterProduction
$waterFormationVolumeFactor
$cumulativeGasInjection
$CO2DensityAtRefDepthAndVirginReservoirConditions

static calcGrv (line 111)

access: public

void calcGrv ($area, $grossthickness, $estimatedRelief, $shapeFactor, $isStructuralStratigraphicTrap, $isSalineAquifer)

$area
$grossthickness
$estimatedRelief
$shapeFactor
$isStructuralStratigraphicTrap
$isSalineAquifer

static calcHydrostatic (line 138)

access: public

void calcHydrostatic ($shallowestdepth)

$shallowestdepth

static calcInitialPorePressureAtCentroid (line 145)

access: public

void calcInitialPorePressureAtCentroid ($porepressure, $meandepth, $shallowestdepth)

$porepressure
$meandepth
$shallowestdepth

static calcLithostaticPresureAtCentroidDepth (line 496)

defined ticket 124 *

defined ticket 124 *

access: public

void calcLithostaticPresureAtCentroidDepth ($lithostaticPressureAtShallowestdepth, $centroidDepth, $shallowestDepth)

$lithostaticPressureAtShallowestdepth
$centroidDepth
$shallowestDepth

static calcMaxCumulativeInjectionPerWellForGas (line 586)

defined ticket 134.

defined ticket 134. *

access: public

void calcMaxCumulativeInjectionPerWellForGas ($MaxCumulativeProductionPerWell, $GasFormationVolumeFactor, $CO2DensityAtReferenceDepth)

$MaxCumulativeProductionPerWell
$GasFormationVolumeFactor
$CO2DensityAtReferenceDepth

static calcMaxCumulativeInjectionPerWellForGasCondensate (line 609)

defined ticket 134.

defined ticket 134. *

access: public

void calcMaxCumulativeInjectionPerWellForGasCondensate ($MaxCumulativeProductionPerWell, $GasShrinkageFactor, $GasFormationVolumeFactor, $CO2DensityAtReferenceDepth)

$MaxCumulativeProductionPerWell
$GasShrinkageFactor
$GasFormationVolumeFactor
$CO2DensityAtReferenceDepth

static calcMaxCumulativeInjectionPerWellForOilAndGas (line 563)

defined ticket 134.

defined ticket 134. *

access: public

void calcMaxCumulativeInjectionPerWellForOilAndGas ($MaxCumulativeProductionPerWell, $OilFormationVolumeFactor, $CO2DensityAtReferenceDepth)

$MaxCumulativeProductionPerWell
$OilFormationVolumeFactor
$CO2DensityAtReferenceDepth

static calcMaxPressure (line 516)

defined ticket 124 *

defined ticket 124 *

access: public

void calcMaxPressure ($lithostaticPressureAtCentroidDepth, $fracturePressureAtCentroidDepth)

$lithostaticPressureAtCentroidDepth
$fracturePressureAtCentroidDepth

static calcMinMlMaxForPVMultCombinedDev (line 635)

access: public

void calcMinMlMaxForPVMultCombinedDev ($numberOfSalineDaughters)

$numberOfSalineDaughters

static calcNumberWellsForHydroCarbonInjectivity (line 617)

access: public

void calcNumberWellsForHydroCarbonInjectivity ($inj, $dur, $hydrocarbonStorageCapacity, $estimatedInjRatePerWell, $maxCumInjPerWell)

$inj
$dur
$hydrocarbonStorageCapacity
$estimatedInjRatePerWell
$maxCumInjPerWell

static calcOilGasStorageCapacity (line 388)

access: public

void calcOilGasStorageCapacity ($cumulativeOilProduction, $oilFormationVolumeFactor, $cumulativeGasProduction, $solutionGasOilRatioAtVirginReservoirConditions, $gasFormationVolumeFactor, $cumulativeWaterProduction, $waterFormationVolumeFactor, $cumulativeWaterInjection, $cumulativeGasInjection, $CO2DensityAtRefDepthAndVirginReservoirConditions)

$cumulativeOilProduction
$oilFormationVolumeFactor
$cumulativeGasProduction
$solutionGasOilRatioAtVirginReservoirConditions
$gasFormationVolumeFactor
$cumulativeWaterProduction
$waterFormationVolumeFactor
$cumulativeWaterInjection
$cumulativeGasInjection
$CO2DensityAtRefDepthAndVirginReservoirConditions

static calcOpenAquiferParentCapacity (line 481)

access: public

void calcOpenAquiferParentCapacity ($poreVolume, $poreSpaceUtilisation, $pc02)

$poreVolume
$poreSpaceUtilisation
$pc02

static calcPoreSpaceUtilisationForTicket124 (line 540)

defined ticket 124.

defined ticket 124. Sometimes you calculate this from standard values, sometimes from this forumla. *

access: public

void calcPoreSpaceUtilisationForTicket124 ($PVMultCombinedDev, $swirr, $volumeSweep)

$PVMultCombinedDev
$swirr
$volumeSweep

static calcPoreVolume (line 131)

access: public

void calcPoreVolume ($grv, $arealsand, $ntg, $porosity)

$grv
$arealsand
$ntg
$porosity

static calcPredictedMigrationVelocity (line 472)

access: public

void calcPredictedMigrationVelocity ($storagePermeability, $pc02, $dip, $krco2, $uco2, $averagePorosity, $swirr)

$storagePermeability
$pc02
$dip
$krco2
$uco2
$averagePorosity
$swirr

static calcPressureCapacity (line 308)

*

*

access: public

void calcPressureCapacity ($porevolume, $formationwatercompress, $rockcompress, $aquifercapacity, $codensity)

$porevolume
$formationwatercompress
$rockcompress
$aquifercapacity
$codensity

static calcPressureForHydroCarbonInjectivity (line 628)

access: public

void calcPressureForHydroCarbonInjectivity ($referenceDepth)

$referenceDepth

static calcPVMultFullyConfined (line 524)

defined ticket 124 *

defined ticket 124 *

access: public

void calcPVMultFullyConfined ($maxPressure, $initalPorePressure, $totalCompressibility)

$maxPressure
$initalPorePressure
$totalCompressibility

static calcPVxEfactor (line 317)

*

*

access: public

void calcPVxEfactor ($storageefficiency, $codensity, $porevolume)

$storageefficiency
$codensity
$porevolume

static calcRockCompressMax (line 278)

access: public

void calcRockCompressMax ($averagePorosityML)

$averagePorosityML

static calcRockCompressMin (line 264)

access: public

void calcRockCompressMin ($averagePorosityML)

$averagePorosityML

static calcRockCompressML (line 271)

access: public

void calcRockCompressML ($averagePorosityML)

$averagePorosityML

static calcScaledFormationArea (line 532)

defined ticket 124 *

defined ticket 124 *

access: public

void calcScaledFormationArea ($formationArea, $PoreSpaceUtilisation, $PVMult_FullyConfined)

$formationArea
$PoreSpaceUtilisation
$PVMult_FullyConfined

static calcTemperatureGradient (line 184)

access: public

void calcTemperatureGradient ($formationtemp, $shallowestdepth)

$formationtemp
$shallowestdepth

static calcThicknessArea (line 124)

access: public

void calcThicknessArea ($grossthickness, $area)

$grossthickness
$area

static calcTotalCompressibility (line 488)

access: public

void calcTotalCompressibility ($rockCompressibility, $formationWaterCompressibility)

$rockCompressibility
$formationWaterCompressibility

static cleanResult (line 327)

Do NOT return NANs or INFS *

Do NOT return NANs or INFS *

access: public

void cleanResult ($in)

$in

static find2NearestPointsInArray (line 35)

access: public

void find2NearestPointsInArray ($array, $desired)

$array
$desired

static getLookUpTable (line 88)

Just gets data from disk and returns it in array

Just gets data from disk and returns it in array 1st key is vertical lines, or X 2nd key is vertical lines, or Y *

access: public

void getLookUpTable ($name)

$name

static getValueFromLookUpTable (line 53)

access: public

void getValueFromLookUpTable ($table, $X, $Y)

$table
$X
$Y

static interpolateBetween2Values (line 47)

Assume $value1idx is less than $value2idx.

Assume $value1idx is less than $value2idx. *

access: public

void interpolateBetween2Values ($value1idx, $value1data, $value2idx, $value2data, $desired)

$value1idx
$value1data
$value2idx
$value2data
$desired

Class Constants
BRINE_DENSITY = 1.026 (line 13)
ESTIMATED_INJECTION_RATE_PER_WELL_MAX = 3.5 (line 29)
ESTIMATED_INJECTION_RATE_PER_WELL_MIN = 0.1 (line 28)
GRAVITY = 9.8067 (line 12)
MAX_ALLOWABLE_STORAGE_PRESSURE_AS_A_FRACTION_OF_FRACTURE_PRESSURE = 0.9 (line 10)
MAX_CUMULATIVE_INJECTION_RATE_PER_WELL_MAX = 20 (line 32)
MAX_CUMULATIVE_INJECTION_RATE_PER_WELL_MIN = 0.75 (line 31)
MIGRATION_VELOCITY_THRESHHOLD = 10 (line 18)
PERMEABILITY_THRESHHOLD = 10 (line 17)
PVMULTCOMBINEDDEV_MAX_ADDITION = 33 (line 26)
PVMULTCOMBINEDDEV_MAX_MULT = 0.0 (line 22)
PVMULTCOMBINEDDEV_MIN_ADDITION = 1 (line 24)
PVMULTCOMBINEDDEV_MIN_MULT = 0.0 (line 20)
PVMULTCOMBINEDDEV_ML_ADDITION = 20.3 (line 25)
PVMULTCOMBINEDDEV_ML_MULT = -0.897 (line 21)
REPRESENTATIVE_FRACTURE_PRESSURE_GRADIENT = 19 (line 15)
REPRESENTATIVE_LITHOSTATIC_PRESSURE_GRADIENT = 20 (line 14)
SALINE_WATER_PRESSURE_GRADIENT = 10.0659 (line 9)
SURFACE_TEMPERATURE = 8 (line 11)