## Contents

1 **Hierarchical Index**
   1.1 Class Hierarchy ................................................. 1

2 **Class Index**
   2.1 Class List .................................................. 3

3 **Class Documentation**
   3.1 **STSAttribute Class Reference** .................................. 7
      3.1.1 Detailed Description ..................................... 8
      3.1.2 Method Documentation .................................... 9
   3.2 **STSAttributeList Class Reference** ............................ 10
      3.2.1 Detailed Description .................................... 11
      3.2.2 Method Documentation .................................... 11
   3.3 **STSAttributeListIterator Class Reference** .................... 11
      3.3.1 Detailed Description ..................................... 12
      3.3.2 Method Documentation .................................... 13
   3.4 **STSAttributeStatistics Class Reference** ...................... 13
      3.4.1 Detailed Description ..................................... 14
      3.4.2 Method Documentation .................................... 15
   3.5 **STSBooleanList Class Reference** ............................... 17
      3.5.1 Detailed Description ..................................... 18
      3.5.2 Method Documentation .................................... 18
   3.6 **STSBooleanListIterator Class Reference** ...................... 19
      3.6.1 Detailed Description ..................................... 19
      3.6.2 Method Documentation .................................... 20
   3.7 **STSCommunitiesSCD Class Reference** ............................ 20
      3.7.1 Detailed Description ..................................... 22
      3.7.2 Method Documentation .................................... 22
   3.8 **STSCommunityDetection Class Reference** ....................... 24
      3.8.1 Detailed Description ..................................... 25
      3.8.2 Method Documentation .................................... 25
   3.9 **STSCo m binedComponents Class Reference** .................... 26
      3.9.1 Detailed Description ..................................... 27
      3.9.2 Method Documentation .................................... 28
   3.10 **STSConnectivity Class Reference** .............................. 29
      3.10.1 Detailed Description ..................................... 30
      3.10.2 Method Documentation .................................... 31
   3.11 **STSContext Class Reference** .................................. 32
      3.11.1 Detailed Description ..................................... 33
3.11.2 Method Documentation .................................................. 33
3.12 STSCSVReader Class Reference ........................................... 35
  3.12.1 Detailed Description .................................................. 37
  3.12.2 Method Documentation ............................................... 37
3.13 STSCSVWriter Class Reference ........................................... 40
  3.13.1 Detailed Description .................................................. 41
  3.13.2 Method Documentation ............................................... 42
3.14 STSDatabase Class Reference ............................................. 43
  3.14.1 Detailed Description .................................................. 45
  3.14.2 Method Documentation ............................................... 45
3.15 STSDatabaseStatistics Class Reference ................................ 46
  3.15.1 Detailed Description .................................................. 47
  3.15.2 Method Documentation ............................................... 47
3.16 STSDefaultExport Class Reference .................................... 48
  3.16.1 Detailed Description .................................................. 50
  3.16.2 Method Documentation ............................................... 50
3.17 STSDisjointCommunities Class Reference ............................ 52
  3.17.1 Detailed Description .................................................. 53
  3.17.2 Method Documentation ............................................... 53
3.18 STSDisjointCommunityDetection Class Reference .................... 54
  3.18.1 Detailed Description .................................................. 56
  3.18.2 Method Documentation ............................................... 56
3.19 STSEdgeData Class Reference ........................................... 58
  3.19.1 Detailed Description .................................................. 59
  3.19.2 Method Documentation ............................................... 59
3.20 STSEdgeExport Class Reference ......................................... 59
  3.20.1 Detailed Description .................................................. 61
  3.20.2 Method Documentation ............................................... 61
3.21 STSEdgeTypeExporter Class Reference ................................ 64
  3.21.1 Detailed Description .................................................. 66
  3.21.2 Method Documentation ............................................... 66
3.22 STSEdgeTypeLoader Class Reference .................................. 68
  3.22.1 Detailed Description .................................................. 70
  3.22.2 Method Documentation ............................................... 70
3.23 STSExportManager Class Reference ................................. 73
  3.23.1 Detailed Description .................................................. 75
  3.23.2 Method Documentation ............................................... 75
3.24 STSGraph Class Reference .............................................. 77
  3.24.1 Detailed Description .................................................. 80
  3.24.2 Method Documentation ............................................... 80
3.25 STSGraphExport Class Reference ......................................................... 95
  3.25.1 Detailed Description ................................................................. 96
  3.25.2 Method Documentation .............................................................. 96

3.26 STSInt32List Class Reference .......................................................... 96
  3.26.1 Detailed Description ................................................................. 98
  3.26.2 Method Documentation .............................................................. 98

3.27 STSInt32ListIterator Class Reference ............................................... 98
  3.27.1 Detailed Description ................................................................. 99
  3.27.2 Method Documentation .............................................................. 99

3.28 STSNodeExport Class Reference ......................................................... 100
  3.28.1 Detailed Description ................................................................. 101
  3.28.2 Method Documentation .............................................................. 102

3.29 STSNodeTypeExporter Class Reference ................................................. 105
  3.29.1 Detailed Description ................................................................. 106
  3.29.2 Method Documentation .............................................................. 107

3.30 STSNodeTypeLoader Class Reference ................................................... 108
  3.30.1 Detailed Description ................................................................. 110
  3.30.2 Method Documentation .............................................................. 110

3.31 STSObjects Class Reference ............................................................. 112
  3.31.1 Detailed Description ................................................................. 114
  3.31.2 Method Documentation .............................................................. 115

3.32 STSObjectsIterator Class Reference .................................................. 119
  3.32.1 Detailed Description ................................................................. 120
  3.32.2 Method Documentation .............................................................. 120

3.33 STSOidList Class Reference .............................................................. 121
  3.33.1 Detailed Description ................................................................. 122
  3.33.2 Method Documentation .............................................................. 122

3.34 STSOidListIterator Class Reference .................................................. 123
  3.34.1 Detailed Description ................................................................. 124
  3.34.2 Method Documentation .............................................................. 124

3.35 STSPlatform Class Reference ............................................................ 125
  3.35.1 Detailed Description ................................................................. 125
  3.35.2 Method Documentation .............................................................. 126

3.36 STSPlatformStatistics Class Reference .............................................. 126
  3.36.1 Detailed Description ................................................................. 127
  3.36.2 Method Documentation .............................................................. 127

3.37 STSQuery Class Reference ............................................................... 128
  3.37.1 Detailed Description ................................................................. 129
  3.37.2 Method Documentation .............................................................. 129

3.38 STSQueryContext Class Reference ..................................................... 130
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.38.1 Detailed Description</td>
<td>131</td>
</tr>
<tr>
<td>3.39 STSQueryStream Class Reference</td>
<td>131</td>
</tr>
<tr>
<td>3.39.1 Detailed Description</td>
<td>132</td>
</tr>
<tr>
<td>3.39.2 Method Documentation</td>
<td>132</td>
</tr>
<tr>
<td>3.40 STSResultSet Class Reference</td>
<td>133</td>
</tr>
<tr>
<td>3.40.1 Detailed Description</td>
<td>134</td>
</tr>
<tr>
<td>3.40.2 Method Documentation</td>
<td>134</td>
</tr>
<tr>
<td>3.41 STSResultSetList Class Reference</td>
<td>136</td>
</tr>
<tr>
<td>3.41.1 Detailed Description</td>
<td>137</td>
</tr>
<tr>
<td>3.41.2 Method Documentation</td>
<td>137</td>
</tr>
<tr>
<td>3.42 STSResultSetListIterator Class Reference</td>
<td>137</td>
</tr>
<tr>
<td>3.42.1 Detailed Description</td>
<td>138</td>
</tr>
<tr>
<td>3.42.2 Method Documentation</td>
<td>139</td>
</tr>
<tr>
<td>3.43 STSRowReader Class Reference</td>
<td>139</td>
</tr>
<tr>
<td>3.43.1 Detailed Description</td>
<td>140</td>
</tr>
<tr>
<td>3.43.2 Method Documentation</td>
<td>140</td>
</tr>
<tr>
<td>3.44 STSRowWriter Class Reference</td>
<td>141</td>
</tr>
<tr>
<td>3.44.1 Detailed Description</td>
<td>142</td>
</tr>
<tr>
<td>3.44.2 Method Documentation</td>
<td>143</td>
</tr>
<tr>
<td>3.45 STSScriptParser Class Reference</td>
<td>143</td>
</tr>
<tr>
<td>3.45.1 Detailed Description</td>
<td>144</td>
</tr>
<tr>
<td>3.45.2 Method Documentation</td>
<td>144</td>
</tr>
<tr>
<td>3.46 STSSession Class Reference</td>
<td>146</td>
</tr>
<tr>
<td>3.46.1 Detailed Description</td>
<td>147</td>
</tr>
<tr>
<td>3.46.2 Method Documentation</td>
<td>147</td>
</tr>
<tr>
<td>3.47 STSShortestPath Class Reference</td>
<td>147</td>
</tr>
<tr>
<td>3.47.1 Detailed Description</td>
<td>149</td>
</tr>
<tr>
<td>3.47.2 Method Documentation</td>
<td>149</td>
</tr>
<tr>
<td>3.48 STSSinglePairShortestPath Class Reference</td>
<td>150</td>
</tr>
<tr>
<td>3.48.1 Detailed Description</td>
<td>152</td>
</tr>
<tr>
<td>3.48.2 Method Documentation</td>
<td>152</td>
</tr>
<tr>
<td>3.49 STSSinglePairShortestPathBFS Class Reference</td>
<td>154</td>
</tr>
<tr>
<td>3.49.1 Detailed Description</td>
<td>156</td>
</tr>
<tr>
<td>3.49.2 Method Documentation</td>
<td>156</td>
</tr>
<tr>
<td>3.50 STSSinglePairShortestPathDijkstra Class Reference</td>
<td>158</td>
</tr>
<tr>
<td>3.50.1 Detailed Description</td>
<td>160</td>
</tr>
<tr>
<td>3.50.2 Method Documentation</td>
<td>160</td>
</tr>
<tr>
<td>3.51 STSSparksee Class Reference</td>
<td>162</td>
</tr>
<tr>
<td>3.51.1 Detailed Description</td>
<td>164</td>
</tr>
<tr>
<td>3.51.2 Method Documentation</td>
<td>164</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
</tr>
<tr>
<td>---------</td>
<td>--------------------------------------------</td>
</tr>
<tr>
<td>3.65</td>
<td>Detailed Description</td>
</tr>
<tr>
<td>3.65</td>
<td>Method Documentation</td>
</tr>
<tr>
<td>3.66</td>
<td>STSTypeList Class Reference</td>
</tr>
<tr>
<td>3.66</td>
<td>Detailed Description</td>
</tr>
<tr>
<td>3.66</td>
<td>Method Documentation</td>
</tr>
<tr>
<td>3.67</td>
<td>STSTypeListIterator Class Reference</td>
</tr>
<tr>
<td>3.67</td>
<td>Detailed Description</td>
</tr>
<tr>
<td>3.67</td>
<td>Method Documentation</td>
</tr>
<tr>
<td>3.68</td>
<td>STSTypeLoader Class Reference</td>
</tr>
<tr>
<td>3.68</td>
<td>Detailed Description</td>
</tr>
<tr>
<td>3.68</td>
<td>Method Documentation</td>
</tr>
<tr>
<td>3.69</td>
<td>STSTypeLoaderEvent Class Reference</td>
</tr>
<tr>
<td>3.69</td>
<td>Detailed Description</td>
</tr>
<tr>
<td>3.69</td>
<td>Method Documentation</td>
</tr>
<tr>
<td>3.70</td>
<td>STSTypeLoaderListener Class Reference</td>
</tr>
<tr>
<td>3.70</td>
<td>Detailed Description</td>
</tr>
<tr>
<td>3.70</td>
<td>Method Documentation</td>
</tr>
<tr>
<td>3.71</td>
<td>STSValue Class Reference</td>
</tr>
<tr>
<td>3.71</td>
<td>Detailed Description</td>
</tr>
<tr>
<td>3.71</td>
<td>Method Documentation</td>
</tr>
<tr>
<td>3.72</td>
<td>STSValueList Class Reference</td>
</tr>
<tr>
<td>3.72</td>
<td>Detailed Description</td>
</tr>
<tr>
<td>3.72</td>
<td>Method Documentation</td>
</tr>
<tr>
<td>3.73</td>
<td>STSValueListIterator Class Reference</td>
</tr>
<tr>
<td>3.73</td>
<td>Detailed Description</td>
</tr>
<tr>
<td>3.73</td>
<td>Method Documentation</td>
</tr>
<tr>
<td>3.74</td>
<td>STSValues Class Reference</td>
</tr>
<tr>
<td>3.74</td>
<td>Detailed Description</td>
</tr>
<tr>
<td>3.74</td>
<td>Method Documentation</td>
</tr>
<tr>
<td>3.75</td>
<td>STSValuesIterator Class Reference</td>
</tr>
<tr>
<td>3.75</td>
<td>Detailed Description</td>
</tr>
<tr>
<td>3.75</td>
<td>Method Documentation</td>
</tr>
<tr>
<td>3.76</td>
<td>STSWeakConnectivity Class Reference</td>
</tr>
<tr>
<td>3.76</td>
<td>Detailed Description</td>
</tr>
<tr>
<td>3.76</td>
<td>Method Documentation</td>
</tr>
<tr>
<td>3.77</td>
<td>STSWeakConnectivityDFS Class Reference</td>
</tr>
<tr>
<td>3.77</td>
<td>Detailed Description</td>
</tr>
<tr>
<td>3.77</td>
<td>Method Documentation</td>
</tr>
<tr>
<td>3.78</td>
<td>SWIGTYPE_p_wchar_t Class Reference</td>
</tr>
</tbody>
</table>
1 Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

- NSObject
- STSAttribute
- STSAttributeList
- STSAttributeListIterator
- STSAttributeStatistics
- STSBooleanList
- STSBooleanListIterator
- STSCommunityDetection
- STSDisjointCommunityDetection
- STSCommunitiesSCD
- STSConnectedComponents
- STSConnectivity
- STSStrongConnectivity
- STSStrongConnectivityGabow
- STSWeakConnectivity
- STSWeakConnectivityDFS
- STSContext
- STSDatabase
- STSDatabaseStatistics
- STSDisjointCommunities
- STSEdgeData
- STSEdgeExport
- STSExportManager
- STSDefaultExport
- STSGraph
- STSGraphExport
- STSInt32List
1.1 Class Hierarchy

STSInt32ListIterator 98
STSNodeExport 100
STSOBJECTS 112
STSOBJECTSIterator 119
STSOidList 121
STSOidListIterator 123
STSPROTOCOL 125
STSPROTOCOLStatistics 126
STSQUERY 128
STSSCOLContext 130
STSSQUERYStream 131
STSSRESULTSet 133
STSSRESULTSetList 136
STSSRESULTSetListIterator 137
STSSROWReader 139
STSCSVReader 35
STSSROWWriter 141
STSCSVWriter 40
STSSCRIPTParser 143
STSSSESSION 146
STSSHORTESTPath 147
STSSHORTESTPath 150
STSSHORTESTPathBFS 154
STSSHORTESTPathDIJKSTRA 158
STSSPARKsee 162
STSSPARKseeConfig 165
STSSPARKseeProperties 176
STSSSTRINGList 179
STSSSTRINGListIterator 181
STSTEXTStream 190
STSTRANSION 192
STSTRANSIONBFS 196
2 Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

**STSTraversalDFS**
Attribute data class 199

**STSType**
203

**STSTypeExporter**
205

**STSEdgeTypeExporter**
64

**STSNodetypeExporter**
105

**STSTypeExporterEvent**
208

**STSTypeExporterListener**
210

**STSTypeList**
211

**STSTypeListIterator**
213

**STSTypeLoader**
215

**STSEdgeTypeLoader**
68

**STSNodetypeLoader**
108

**STSTypeLoaderEvent**
219

**STSTypeLoaderListener**
221

**STSTuple**
223

**STSTupleList**
231

**STSTupleListIterator**
233

**STSTupleListIterator**
235

**STSTupleList**
237

**SWIGTYPE_p_wchar_t**
246

Generated on Tue Jun 2 2015 15:49:36 for Sparksee by Doxygen
<table>
<thead>
<tr>
<th>Class Name</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>STSBooleanListIterator</td>
<td>BooleanList iterator class</td>
<td>19</td>
</tr>
<tr>
<td>STSCommunitiesSCD</td>
<td>CommunitiesSCD class</td>
<td>20</td>
</tr>
<tr>
<td>STSCommunityDetection</td>
<td>CommunityDetection class</td>
<td>24</td>
</tr>
<tr>
<td>STSConnectedComponents</td>
<td>ConnectedComponents class</td>
<td>26</td>
</tr>
<tr>
<td>STSConnectivity</td>
<td>Connectivity class</td>
<td>29</td>
</tr>
<tr>
<td>STSContext</td>
<td>Context class</td>
<td>32</td>
</tr>
<tr>
<td>STSCSVReader</td>
<td>CSVReader interface</td>
<td>35</td>
</tr>
<tr>
<td>STSCSVWriter</td>
<td>CSVWriter interface</td>
<td>40</td>
</tr>
<tr>
<td>STSDatabase</td>
<td>Database class</td>
<td>43</td>
</tr>
<tr>
<td>STSDatabaseStatistics</td>
<td>Database statistics</td>
<td>46</td>
</tr>
<tr>
<td>STSDefaultExport</td>
<td>Default implementation for ExportManager class</td>
<td>48</td>
</tr>
<tr>
<td>STSDisjointCommunities</td>
<td>DisjointCommunities class</td>
<td>52</td>
</tr>
<tr>
<td>STSDisjointCommunityDetection</td>
<td>DisjointCommunityDetection class</td>
<td>54</td>
</tr>
<tr>
<td>STSEdgeData</td>
<td>Edge data class</td>
<td>58</td>
</tr>
<tr>
<td>STSEdgeExport</td>
<td>Stores edge exporting values</td>
<td>59</td>
</tr>
<tr>
<td>STSEdgeTypeExporter</td>
<td>EdgeTypeExporter class</td>
<td>64</td>
</tr>
<tr>
<td>STSEdgeTypeLoader</td>
<td>EdgeTypeLoader class</td>
<td>68</td>
</tr>
<tr>
<td>STSExportManager</td>
<td>Defines how to export a graph to an external format</td>
<td>73</td>
</tr>
<tr>
<td>STSGraph</td>
<td>Graph class</td>
<td>77</td>
</tr>
<tr>
<td>STSGraphExport</td>
<td>Stores the graph exporting values</td>
<td>95</td>
</tr>
<tr>
<td>STSInt32List</td>
<td>Sparksee 32-bit signed integer list</td>
<td>96</td>
</tr>
<tr>
<td>Class Name</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>STSInt32ListIterator</td>
<td>Int32List iterator class</td>
<td>98</td>
</tr>
<tr>
<td>STSNodeExport</td>
<td>Stores the node exporting values</td>
<td>100</td>
</tr>
<tr>
<td>STSNodeTypeExporter</td>
<td>NodeTypeExporter class</td>
<td>105</td>
</tr>
<tr>
<td>STSNodeTypeLoader</td>
<td>NodeTypeLoader class</td>
<td>108</td>
</tr>
<tr>
<td>STSObjects</td>
<td>Object identifier set class</td>
<td>112</td>
</tr>
<tr>
<td>STSObjectsIterator</td>
<td>ObjectsIterator class</td>
<td>119</td>
</tr>
<tr>
<td>STSOidList</td>
<td>Sparksee object identifier list</td>
<td>121</td>
</tr>
<tr>
<td>STSOidListIterator</td>
<td>OIDList iterator class</td>
<td>123</td>
</tr>
<tr>
<td>STSPlatform</td>
<td>Platform class</td>
<td>125</td>
</tr>
<tr>
<td>STSPlatformStatistics</td>
<td>Platform data and statistics</td>
<td>126</td>
</tr>
<tr>
<td>STSQuery</td>
<td>Query class</td>
<td>128</td>
</tr>
<tr>
<td>STSQueryContext</td>
<td>Query context interface</td>
<td>130</td>
</tr>
<tr>
<td>STSQueryStream</td>
<td>Query stream interface</td>
<td>131</td>
</tr>
<tr>
<td>STSResultSet</td>
<td>ResultSet class</td>
<td>133</td>
</tr>
<tr>
<td>STSResultSetList</td>
<td>ResultSet list</td>
<td>136</td>
</tr>
<tr>
<td>STSResultSetListIterator</td>
<td>ResultSetList iterator class</td>
<td>137</td>
</tr>
<tr>
<td>STSRowReader</td>
<td>RowReader interface</td>
<td>139</td>
</tr>
<tr>
<td>STSRowWriter</td>
<td>RowWriter interface</td>
<td>141</td>
</tr>
<tr>
<td>STSScriptParser</td>
<td>ScriptParser</td>
<td>143</td>
</tr>
<tr>
<td>STSSession</td>
<td>Session class</td>
<td>146</td>
</tr>
<tr>
<td>STSShortestPath</td>
<td>ShortestPath class</td>
<td>147</td>
</tr>
</tbody>
</table>
2.1 Class List

STSSinglePairShortestPath
  SinglePairShortestPath class 150

STSSinglePairShortestPathBFS
  SinglePairShortestPathBFS class 154

STSSinglePairShortestPathDijkstra
  SinglePairShortestPathDijkstra class 158

STSSparksee
  Sparksee class 162

STSSparkseeConfig
  Sparksee configuration class 165

STSSparkseeProperties
  Sparksee properties file 176

STSStringList
  String list 179

STSStringListIterator
  StringList iterator class 181

STSStrongConnectivity
  StrongConnectivity class 183

STSStrongConnectivityGabow
  This class can be used to solve the problem of finding strongly connected components in a
directed graph 186

STSTextStream
  TextStream class 190

STSTraversal
  Traversal class 192

STSTraversalBFS
  Breadth-First Search implementation of Traversal 196

STSTraversalDFS
  Depth-First Search (DFS) implementation of Traversal 199

STSType
  Type data class 203

STSTypeExporter
  Base TypeExporter class 205

STSTypeExporterEvent
  Provides information about the progress of an TypeExproter instance 208

STSTypeExporterListener
  Interface to be implemented to receive TypeExporterEvent events from a TypeExporter 210

STSTypeList
  Sparksee type identifier list 211

STSTypeListIterator
  TypeList iterator class 213
3 Class Documentation

3.1 STSAttribute Class Reference

Attribute data class.

Inheritance diagram for STSAttribute:
Collaboration diagram for STSAttribute:

![Collaboration Diagram]

### Instance Methods

- `(int) - getId`
  
  
  Gets the Sparksee attribute identifier.

- `(int) - getTypeId`
  
  Gets the Sparksee type identifier.

- `(NSString *) - getName`
  
  Gets the unique attribute name.

- `(enum STSDataType) - getDataType`
  
  Gets the data type.

- `(long long) - getSize`
  
  Gets the number of different values.

- `(long long) - getCount`
  
  Gets the number of non-NULL values.

- `(enum STSAttributeKind) - getKind`
  
  Gets the attribute kind.

- `(BOOL) - isSessionAttribute`
  
  Check if it's a session attribute or a persistent one.

### Class Methods

- `(int) + getInvalidAttribute`
  
  Invalid attribute identifier constant.

### 3.1.1 Detailed Description

Attribute data class.

It contains information about an attribute.

**Author**

Sparsity Technologies [http://www.sparsity-technologies.com](http://www.sparsity-technologies.com)
3.1 Method Documentation

3.1.2 - (long long) getCount

Gets the number of non-NULL values.

Returns
   The number of non-NULL values.

3.1.2.2 - (enum STSDataType) getDataType

Gets the data type.

Returns
   The DataType.

3.1.2.3 - (int) getId

Gets the Sparksee attribute identifier.

Returns
   The Sparksee attribute identifier.

3.1.2.4 - (enum STSAttributeKind) getKind

Gets the attribute kind.

Returns
   The AttributeKind.

3.1.2.5 - (NSString *) getName

Gets the unique attribute name.

Returns
   The unique attribute name.

3.1.2.6 - (long long) getSize

Gets the number of different values.

Returns
   The number of different values.

3.1.2.7 - (int) getTypeId

Gets the Sparksee type identifier.

Returns
   The Sparksee type identifier.
3.1.2.8 - (BOOL) isSessionAttribute

Check if it's a session attribute or a persistent one.

Returns

True if it's a session attribute, or false otherwise.

The documentation for this class was generated from the following file:

- Sparksee.h

3.2 STSAttributeList Class Reference

Sparksee attribute identifier list.

Inheritance diagram for STSAttributeList:

Collaboration diagram for STSAttributeList:

Instance Methods

- (int) - count
  
  Number of elements in the list.
- (id) - init
3.3 STSAttributeListIterator Class Reference

Constructor.
• (void) - add:

An attribute identifier at the end of the list.
• (void) - clear

Clears the list.
• (id) - initWithArray:

Creates a new AttributeList instance from the given array.
• (id) - initWithNSEnumerator:

Creates a new AttributeList instance from the given NSEnumerator.
• (STSAttributeListIterator *) - iterator

Gets a new AttributeListIterator.

3.2.1 Detailed Description

Sparksee attribute identifier list.
It stores a Sparksee attribute identifier list.
Use AttributeListIterator to access all elements into this collection.

Author

Sparsity Technologies http://www.sparsity-technologies.com

3.2.2 Method Documentation

3.2.2.1 - (void) add: (int) attr

Adds a Sparksee attribute identifier at the end of the list.

Parameters

attr [in] Sparksee attribute identifier.

3.2.2.2 - (int) count

Number of elements in the list.

Returns

Number of elements in the list.

3.2.2.3 - (id) init

Constructor.

This creates an empty list.
The documentation for this class was generated from the following file:

• Sparksee.h

3.3 STSAttributeListIterator Class Reference

AttributeList iterator class.
Inheritance diagram for STSAttributeListIterator:

![Inheritance Diagram](image)

Collaboration diagram for STSAttributeListIterator:

![Collaboration Diagram](image)

### Instance Methods

- **(int) - next**
  
  *Moves to the next element.*

- **(BOOL) - hasNext**
  
  *Gets if there are more elements.*

### 3.3.1 Detailed Description

AttributeList iterator class.

Iterator to traverse all the Sparksee attribute identifier into a AttributeList instance.
3.4 STSAttributeStatistics Class Reference

Attribute statistics class.

Inheritance diagram for STSAttributeStatistics:

```
   NSObject
      
STSAttributeStatistics
```

Author

Sparsity Technologies [http://www.sparsity-technologies.com](http://www.sparsity-technologies.com)

3.3.2 Method Documentation

3.3.2.1 - (BOOL) hasNext

Gets if there are more elements.

Returns

TRUE if there are more elements, FALSE otherwise.

3.3.2.2 - (int) next

Moves to the next element.

Returns

The next element.

The documentation for this class was generated from the following file:

- Sparksee.h
Collaboration diagram for STSAttributeStatistics:

```
NSObject
    STSAttributeStatistics
```

**Instance Methods**

- (long long) - getTotal
  Gets the number of objects with a non-NULL Value (BASIC statistic).
- (long long) - getNull
  Gets the number of objects NULL a Value (BASIC statistics).
- (long long) - getDistinct
  Gets the number of distinct values (BASIC statistics).
- (STSValue *) - getMin
  Gets the minimum existing value (BASIC statistics).
- (STSValue *) - getMax
  Gets the maximum existing value (BASIC statistics).
- (int) - getMaxLengthString
  Gets the maximum length.
- (int) - getMinLengthString
  Gets the minimum length.
- (double) - getAvgLengthString
  Gets the average length.
- (STSValue *) - getMode
  Gets the mode.
- (long long) - getModeCount
  Gets the number of objects with a Value equal to the mode.
- (double) - getMean
  Gets the mean or average.
- (double) - getVariance
  Gets the variance.
- (double) - getMedian
  Gets the median.

### 3.4.1 Detailed Description

Attribute statistics class.

It contains statistic data about an attribute.

Some fields are valid just for numerical attributes and others just for string attributes. Also, some statistics are considered BASIC because computing them do not require to traverse all the different values of the attribute. For
each getter method the documentation tells if the statistic is BASIC or not. See the Graph class method getAttributeStatistics or check out the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies [http://www.sparsity-technologies.com](http://www.sparsity-technologies.com)

3.4.2 Method Documentation

3.4.2.1 - (double) getAvgLengthString

Gets the average length.

If the attribute is not an string attribute, it just returns 0.

Returns

The average length.

3.4.2.2 - (long long) getDistinct

Gets the number of distinct values (BASIC statistics).

Returns

The number of distinct values.

3.4.2.3 - (STSValue *) getMax

Gets the maximum existing value (BASIC statistics).

Returns

The maximum existing value.

3.4.2.4 - (int) getMaxLengthString

Gets the maximum length.

If the attribute is not an string attribute, it just returns 0.

Returns

The maximum length.

3.4.2.5 - (double) getMean

Gets the mean or average.

Mean or average: Sum of all Values divided by the number of observations.

It is computed just for numerical attributes.

Returns

The mean.
3.4.2.6 - (double) getMedian

Gets the median.
Median: Middle value that separates the higher half from the lower.
If \( a < b < c \), then the median of the list \([a, b, c]\) is \( b \), and if \( a < b < c < d \), then the median of the list \([a, b, c, d]\) is the mean of \( b \) and \( c \), i.e. it is \( (b + c)/2 \)
It is computed just for numerical attributes.

Returns
The median.

3.4.2.7 - (STSValue*) getMin

Gets the minimum existing value (BASIC statistics).

Returns
The minimum existing value.

3.4.2.8 - (int) getMinLengthString

Gets the minimum length.
If the attribute is not a string attribute, it just returns 0.

Returns
The minimum length.

3.4.2.9 - (STSValue*) getMode

Gets the mode.
Mode: Most frequent Value.

Returns
The mode.

3.4.2.10 - (long long) getModeCount

Gets the number of objects with a Value equal to the mode.

Returns
The number of objects with a Value equal to the mode.

3.4.2.11 - (long long) getNull

Gets the number of objects NULL a Value (BASIC statistics).

Returns
The number of objects NULL a Value.
3.4.2.12 - (long long) getTotal

Gets the number of objects with a non-NULL Value (BASIC statistic).

Returns

The number of objects with a non-NULL Value.

3.4.2.13 - (double) getVariance

Gets the variance.

It is computed just for numerical attributes.

Returns

The variance.

The documentation for this class was generated from the following file:

- Sparksee.h

3.5 STSBooleanList Class Reference

Boolean list.

Inheritance diagram for STSBooleanList:

```
  NSObject
    
  STSBooleanList
```

Collaboration diagram for STSBooleanList:
Instance Methods

- **(int) - count**
  
  *Number of elements in the list.*

- **(id) - init**
  
  *Constructor.*

- **(void) - add:**
  
  *Adds a Boolean at the end of the list.*

- **(void) - clear**
  
  *Clears the list.*

- **(id) - initWithArray:**
  
  *Creates a new BooleanList instance from the given array.*

- **(id) - initWithNSEnumerator:**
  
  *Creates a new BooleanList instance from the given NSEnumerator.*

- **(STSBooleanListIterator *) - iterator**
  
  *Gets a new BooleanListIterator.*

3.5.1 Detailed Description

Boolean list.

It stores a Boolean list.

Use BooleanListIterator to access all elements into this collection.

Author

Sparsity Technologies [http://www.sparsity-technologies.com](http://www.sparsity-technologies.com)

3.5.2 Method Documentation

3.5.2.1 - (void) add: (BOOL) value

Adds a Boolean at the end of the list.

**Parameters**

<table>
<thead>
<tr>
<th>value</th>
<th>[in] Boolean</th>
</tr>
</thead>
</table>

3.5.2.2 - (int) count

Number of elements in the list.

Returns

*Number of elements in the list.*

3.5.2.3 - (id) init

Constructor.

This creates an empty list.

The documentation for this class was generated from the following file:

- Sparksee.h
3.6 STSBooleanListIterator Class Reference

BooleanList iterator class.

Inheritance diagram for STSBooleanListIterator:

```
 NSObject
  ^
 STSBooleanListIterator
```

Collaboration diagram for STSBooleanListIterator:

```
 NSObject
  ^
     ^
     |    
 STSBooleanList
     |    
     v
 theParent
     |
 STSBooleanListIterator
```

Instance Methods

- (BOOL) - next
  Moves to the next element.
- (BOOL) - hasNext
  Gets if there are more elements.

3.6.1 Detailed Description

BooleanList iterator class.

Iterator to traverse all the strings into a BooleanList instance.
3.7 STSCommunitiesSCD Class Reference

Author
Sparsity Technologies http://www.sparsity-technologies.com

3.6.2 Method Documentation

3.6.2.1 - (BOOL) hasNext

Gets if there are more elements.

Returns
TRUE if there are more elements, FALSE otherwise.

3.6.2.2 - (BOOL) next

Moves to the next element.

Returns
The next element.

The documentation for this class was generated from the following file:

• Sparksee.h

3.7 STSCommunitiesSCD Class Reference

CommunitiesSCD class.

Inheritance diagram for STSCommunitiesSCD:

```
 NSObject
    ^
   |  
 STSCommunityDetection
    |
   |  
 STSDisjointCommunityDetection
    |  
   |  
 STSCommunitiesSCD
```
Collaboration diagram for STSCommunitiesSCD:

```
NSObject

STSCommunityDetection

STSDisjointCommunityDetection

STSCommunitiesSCD
```

Instance Methods

- (id) - initWithSession:
  Creates a new instance of CommunitiesSCD.
- (void) - setLookAhead:
  Sets the size of the lookahead iterations to look.
- (void) - run
  Executes the algorithm.
- (void) - addEdgeType:
  Allows connectivity through edges of the given type.
- (void) - addAllEdgeTypes
  Allows connectivity through all edge types of the graph.
- (STSDisjointCommunities *) - getCommunities
  Returns the results generated by the execution of the algorithm.
- (void) - setMaterializedAttribute:
  Creates a new common attribute type for all node types in the graph in order to store, persistently, the results related to the disjoint communities found while executing this algorithm.
- (void) - addNodeType:
  Allows connectivity through nodes of the given type.
- (void) - addAllNodeTypes
  Allows connectivity through all node types of the graph.
- (void) - excludeNodes:
  Set which nodes can’t be used.
- (void) - excludeEdges:
  Set which edges can’t be used.
- (void) - close
3.7 STSCommunitiesSCD Class Reference

Closes the CommunityDetection instance.

- (BOOL) - isClosed
  Check if the CommunityDetection instance is closed.

3.7.1 Detailed Description

CommunitiesSCD class.

Implementation of the community detection algorithm "Scalable Community Detection" based on the paper "High quality, scalable and parallel community detection for large real graphs" by Arnau Prat-Perez, David Dominguez-Sal, Josep-Lluís Larriba-Pey - WWW 2014.

The purpose of this algorithm is to find disjoint communities in an undirected graph or in a directed graph which will be considered as an undirected one.

It is possible to set some restrictions after constructing a new instance of this class and before running it in order to limit the results.

After the execution, we can retrieve the results stored in an instance of the DisjointCommunities class using the getCommunities method.

Check out the 'Algorithms' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies [http://www.sparsity-technologies.com](http://www.sparsity-technologies.com)

3.7.2 Method Documentation

3.7.2.1 - (void) addAllEdgeTypes

Allows connectivity through all edge types of the graph.

The edges can be used in Any direction.

3.7.2.2 - (void) addEdgeType: (int) type

Allows connectivity through edges of the given type.

The edges can be used in Any direction.

Parameters

| type | [in] Edge type. |

3.7.2.3 - (void) addNodeType: (int) type

Allows connectivity through nodes of the given type.

Parameters

| type | null |

3.7.2.4 - (void) close

Closes the CommunityDetection instance.

It must be called to ensure the integrity of all data.

3.7.2.5 - (void) excludeEdges: (STSOBJECTS) edges

Set which edges can’t be used.
This will replace any previously specified set of excluded edges. Should only be used to exclude the usage of specific edges from allowed edge types because it’s less efficient than not allowing an edge type.

Parameters

- `edges` [in] A set of edge identifiers that must be kept intact until the destruction of the class.

3.7.2.6 - (void) excludeNodes: (STSObjects *) nodes

Set which nodes can’t be used.

This will replace any previously specified set of excluded nodes. Should only be used to exclude the usage of specific nodes from allowed node types because it’s less efficient than not allowing a node type.

Parameters

- `nodes` [in] A set of node identifiers that must be kept intact until the destruction of the class.

3.7.2.7 - (STSDisjointCommunities *) getCommunities

Returns the results generated by the execution of the algorithm.

These results contain information related to the disjoint communities found as the number of different components, the set of nodes contained in each component or many other data.

Returns

- Returns an instance of the class DisjointCommunities which contain information related to the disjoint communities found.

3.7.2.8 - (id) initWithSession: (STSSession *) session

Creates a new instance of CommunitiesSCD.

After creating this instance is required to indicate the set of edge types and the set of node types which will be navigated through while traversing the graph in order to find the communities.

Parameters

- `session` [in] Session to get the graph from and calculate the communities

3.7.2.9 - (void) setLookAhead: (int) lookahead

Sets the size of the lookahead iterations to look.

Parameters

- `lookahead` [in] Number of iterations. It must be positive or zero.

3.7.2.10 - (void) setMaterializedAttribute: (NSString *) attributeName

Creates a new common attribute type for all node types in the graph in order to store, persistently, the results related to the disjoint communities found while executing this algorithm.

Whenever the user wants to retrieve the results, even when the graph has been closed and opened again, it is only necessary to create a new instance of the class DisjointCommunities indicating the graph and the name of the common attribute type which stores the results. This instance will have all the information related to the disjoint communities found in the moment of the execution of the algorithm that stored this data.

It is possible to run the algorithm without specifying this parameter in order to avoid materializing the results of the execution.
3.8 STSClCommunityDetection Class Reference

Parameters

| attributeName | [in] The name of the common attribute type for all node types in the graph which will store persistently the results generated by the execution of the algorithm. |

The documentation for this class was generated from the following file:

- Sparksee.h

3.8 STSClCommunityDetection Class Reference

CommunityDetection class.

Inheritance diagram for STSClCommunityDetection:

```
NSObject
  ↓
STSClCommunityDetection
  ↓
STSDisjointCommunityDetection
  ↓
STSClCommunitiesSCD
```

Collaboration diagram for STSClCommunityDetection:
3.8 STSCommunityDetection Class Reference

Instance Methods

- (void) - addNodeType:
  Allows connectivity through nodes of the given type.

- (void) - addAllNodeTypes
  Allows connectivity through all node types of the graph.

- (void) - excludeNodes:
  Set which nodes can't be used.

- (void) - excludeEdges:
  Set which edges can't be used.

- (void) - run
  Runs the algorithm in order to find the connected components.

- (void) - close
  Closes the CommunityDetection instance.

- (BOOL) - isClosed
  Check if the CommunityDetection instance is closed.

3.8.1 Detailed Description

CommunityDetection class.

Any class implementing this abstract class can be used to solve a problem related to graph connectivity as finding the strongly connected components, finding the weakly connected components.

Check out the 'Algorithms' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com

3.8.2 Method Documentation

3.8.2.1 - (void) addNodeType: (int) type

Allows connectivity through nodes of the given type.

Parameters

| type | null |

3.8.2.2 - (void) close

Closes the CommunityDetection instance.

It must be called to ensure the integrity of all data.

3.8.2.3 - (void) excludeEdges: (STSObjects *) edges

Set which edges can't be used.

This will replace any previously specified set of excluded edges. Should only be used to exclude the usage of specific edges from allowed edge types because it's less efficient than not allowing an edge type.

Parameters

| edges | [in] A set of edge identifiers that must be kept intact until the destruction of the class. |
3.8.2.4  - (void) excludeNodes: (STSOBJECTS * ) nodes

Set which nodes can’t be used.
This will replace any previously specified set of excluded nodes. Should only be used to exclude the usage of
specific nodes from allowed node types because it’s less efficient than not allowing a node type.

Parameters

| nodes [in] A set of node identifiers that must be kept intact until the destruction of the class. |

3.8.2.5  - (void) run

Runs the algorithm in order to find the connected components.
This method can be called only once.
Implemented in STSCommunitiesSCD, and STSDisjointCommunityDetection.

The documentation for this class was generated from the following file:

• Sparksee.h

3.9  STSClusterComponents Class Reference

ClusterComponents class.

Inheritance diagram for STSClusterComponents:
Collaboration diagram for STSConnectedComponents:

```
  STSConnectedComponents
  ^
  |  NSObject
```

**Instance Methods**

- `(id)` - `initWithS:materializedattribute:`
  Creates a new instance of ConnectedComponents.

- `(long long)` - `getConnectedComponent`:
  Returns the connected component where the given node belongs to.

- `(long long)` - `getCount`:
  Returns the number of connected components found in the graph.

- `(STSObjects *)` - `getNodes`:
  Returns the collection of nodes contained in the given connected component.

- `(long long)` - `getSize`:
  Returns the number of nodes contained in the given connected component.

- `(void)` - `close`:
  Closes the ConnectedComponents instance.

- `(BOOL)` - `isClosed`:
  Check if the ConnectedComponents instance is closed.

### 3.9.1 Detailed Description

**ConnectedComponents class.**

This class contains the results processed on a Connectivity algorithm. These results contain information related to the connected components found. We must consider that each connected component has a number in order to identify it. These number identifiers are values from 0 to N-1, where N is the number of different connected components found.

When executing any implementation of the Connectivity, it is possible to indicate whether the results of the execution must be stored persistently using the class Connectivity setMaterializedAttribute method. In case the results are set to be materialized, users can retrieve this data whenever they want, even if the graph has been closed and opened again, just by creating a new instance of this class.

Check out the 'Algorithms' section in the SPARKSEE User Manual for more details on this.

**Author**

Sparsity Technologies [http://www.sparsity-technologies.com](http://www.sparsity-technologies.com)
3.9.2 Method Documentation

3.9.2.1 - (void) close

Closes the ConnectedComponents instance. It must be called to ensure the integrity of all data.

3.9.2.2 - (long long) getConnectedComponent: (long long) idNode

Returns the connected component where the given node belongs to.

Parameters

| idNode | [in] The node identifier for which the connected component identifier where it belongs will be returned. |

Returns

The connected component identifier where the given node identifier belongs to.

3.9.2.3 - (long long) getCount

Returns the number of connected components found in the graph.

Returns

The number of connected components found in the graph.

3.9.2.4 - (STSObjects *) getNodes: (long long) idConnectedComponent

Returns the collection of nodes contained in the given connected component.

Parameters

| idConnectedComponent | The connected component for which the collection of nodes contained in it will be returned. |

Returns

The collection of node identifiers contained in the given connected component.

3.9.2.5 - (long long) getSize: (long long) idConnectedComponent

Returns the number of nodes contained in the given connected component.

Parameters

| idConnectedComponent | The connected component for which the number of nodes contained in it will be returned. |

Returns

The number of nodes contained in the given connected component.

3.9.2.6 - (id) initWithS: (STSSession *) s materializedattribute:(NSString *) materializedattribute

Creates a new instance of ConnectedComponents.
This constructor method can only be called when a previous execution of any implementation of the Connectivity class has materialized the results in a common attribute type for all the nodes in the graph. For further information about materializing the results processed on any Connectivity execution see the documentation of the Connectivity-::SetMaterializedAttribute method.

### Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>s</code></td>
<td>[in] Session to get the graph Graph on which the information will be retrieved just by getting the values contained in the given common attribute type for all the nodes in the graph and processing them.</td>
</tr>
<tr>
<td><code>materializedat-tribute</code></td>
<td>[in] The common attribute type for all the nodes in the graph where data will be retrieved in order to process the results related to the connected components found in the graph.</td>
</tr>
</tbody>
</table>

The documentation for this class was generated from the following file:

- Sparksee.h

# 3.10 STSConnectivity Class Reference

Connectivity class.

Inheritance diagram for STSConnectivity:
Collaboration diagram for STSConnectivity:

```
 NSObject

 STSConnectivity
```

**Instance Methods**

- (void) - `addNodeType`:
  Allows connectivity through nodes of the given type.
- (void) - `addAllNodeTypes`:
  Allows connectivity through all node types of the graph.
- (void) - `excludeNodes`:
  Set which nodes can’t be used.
- (void) - `excludeEdges`:
  Set which edges can’t be used.
- (void) - `getConnectedComponents`:
  Return the results generated by the execution of the algorithm.
- (void) - `run`:
  Runs the algorithm in order to find the connected components.
- (void) - `setMaterializedAttribute`:
  Creates a new common attribute type for all node types in the graph in order to store, persistently, the results related to the connected components found while executing this algorithm.
- (void) - `close`:
  Closes the Connectivity instance.
- (BOOL) - `isClosed`:
  Check if the Connectivity instance is closed.

**3.10.1 Detailed Description**

Connectivity class.

Any class implementing this abstract class can be used to solve a problem related to graph connectivity as finding the strongly connected components, finding the weakly connected components.

Check out the 'Algorithms' section in the SPARKSEE User Manual for more details on this.

**Author**

Sparsity Technologies [http://www.sparsity-technologies.com](http://www.sparsity-technologies.com)
3.10.2 Method Documentation

3.10.2.1 - (void) addNodeType: (int) t

Allows connectivity through nodes of the given type.

Parameters

| t | null |

3.10.2.2 - (void) close

Closes the Connectivity instance.

It must be called to ensure the integrity of all data.

3.10.2.3 - (void) excludeEdges: (STSObjects *) edges

Set which edges can’t be used.

This will replace any previously specified set of excluded edges. Should only be used to exclude the usage of specific edges from allowed edge types because it’s less efficient than not allowing an edge type.

Parameters

| edges | [in] A set of edge identifiers that must be kept intact until the destruction of the class. |

3.10.2.4 - (void) excludeNodes: (STSObjects *) nodes

Set which nodes can’t be used.

This will replace any previously specified set of excluded nodes. Should only be used to exclude the usage of specific nodes from allowed node types because it’s less efficient than not allowing a node type.

Parameters

| nodes | [in] A set of node identifiers that must be kept intact until the destruction of the class. |

3.10.2.5 - (STSConnectedComponents *) getConnectedComponents

Returns the results generated by the execution of the algorithm.

These results contain information related to the connected components found as the number of different components, the set of nodes contained in each component or many other data.

Returns

Returns an instance of the class ConnectedComponents which contain information related to the connected components found.

3.10.2.6 - (void) run

Runs the algorithm in order to find the connected components.

This method can be called only once.

Implemented in STSStrongConnectivityGabow, and STSWeakConnectivityDFS.

3.10.2.7 - (void) setMaterializedAttribute: (NSString *) attributeName

Creates a new common attribute type for all node types in the graph in order to store, persistently, the results related to the connected components found while executing this algorithm.
Whenever the user wants to retrieve the results, even when the graph has been closed and opened again, it is only necessary to create a new instance of the class ConnectedComponents indicating the graph and the name of the common attribute type which stores the results. This instance will have all the information related to the connected components found in the moment of the execution of the algorithm that stored this data.

It is possible to run the algorithm without specifying this parameter in order to avoid materializing the results of the execution.

Parameters

| attributeName | [in] The name of the common attribute type for all node types in the graph which will store persistently the results generated by the execution of the algorithm. |

The documentation for this class was generated from the following file:

- Sparksee.h

3.11 STSContext Class Reference

Context class.

Inheritance diagram for STSContext:

![Inheritance Diagram](image)

Collaboration diagram for STSContext:

![Collaboration Diagram](image)
Instance Methods

- (void) - addEdgeType:d:
  Allows for traversing edges of the given type.
- (void) - addAllEdgeTypes:
  Allows for traversing all edge types of the graph.
- (void) - addNodeType:
  Allows for traversing nodes of the given type.
- (void) - addAllNodeTypes
  Allows for traversing all node types of the graph.
- (void) - excludeNodes:
  Set which nodes can’t be used.
- (void) - excludeEdges:
  Set which edges can’t be used.
- (STSObjects *) - compute
  Gets the resulting collection of nodes.
- (void) - setMaximumHops:include:
  Sets the maximum hops restriction.
- (id) - initWithSession:node:
  Creates a new instance.
- (void) - close
  Closes the Context instance.
- (BOOL) - isClosed
  Check if the Context instance is closed.

Class Methods

- (STSObjects *) + computeWithArguments:node:nodeTypes:edgeTypes:dir:maxhops:include:
  Helper method to easily compute a context from a node.

3.11.1 Detailed Description

Context class.

It provides a very similar functionality than the Traversal classes. The main difference is Context returns a resulting collection whereas Traversal provides an iterator behaviour.

Check out the 'Algorithms' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com

3.11.2 Method Documentation

3.11.2.1 - (void) addAllEdgeTypes: (enum STSEdgesDirection) d

Allows for traversing all edge types of the graph.

Parameters

| d | [in] Edge direction. |
3.11.2.2 - (void) addEdgeType: (int) t d:(enum STSEdgesDirection) d

Allows for traversing edges of the given type.

Parameters:

| t  | [in] Edge type. |
| d  | [in] Edge direction. |

3.11.2.3 - (void) addNodeType: (int) t

Allows for traversing nodes of the given type.

Parameters:

| t  | null |

3.11.2.4 - (void) close

Closes the Context instance. It must be called to ensure the integrity of all data.

3.11.2.5 - (STSObjects ∗ ) compute

Gets the resulting collection of nodes.

Returns:

The resulting collection of nodes.

3.11.2.6 + (STSObjects ∗ ) computeWithArguments: (STSSession ∗ ) session node:(long long) node

nodeTypes:(STSTypeList ∗ ) nodeTypes edgeTypes:(STSTypeList ∗ ) edgeTypes dir:(enum STSEdgesDirection) dir
maxhops:(int) maxhops include:(BOOL) include

Helper method to easily compute a context from a node.

Parameters:

| session | [in] Session to get the graph from and perform operation. |
| node    | [in] Node to start traversal from. |
| nodeTypes | [in] Allowed node type list. NULL means all node types are allowed. |
| edgeTypes | [in] Allowed edge type list. NULL means all edge types are allowed. |
| dir     | [in] Allowed direction for the allowed edge types. |
| maxhops | [in] The maximum hops restriction. It must be positive or zero. Zero, the default value, means unlimited. |
| include | [in] If TRUE, the resulting collection will include those nodes at distance less or equal than the given one, otherwise it will just include those nodes at distance equal than the given one. This parameter just makes sense if maxhops is different from 0; in that case it includes all nodes no matters the distance. |

Returns:

Returns an Objects with the computed context of a node.

3.11.2.7 - (void) excludeEdges: (STSObjects ∗ ) edges

Set which edges can’t be used.
This will replace any previously specified set of excluded edges. Should only be used to exclude the usage of specific edges from allowed edge types because it's less efficient than not allowing an edge type.

**Parameters**

| edges | [in] A set of edge identifiers that must be kept intact until the destruction of the class. |

3.11.2.8 - (void) excludeNodes: (STSObjects *) nodes

Set which nodes can't be used.

This will replace any previously specified set of excluded nodes. Should only be used to exclude the usage of specific nodes from allowed node types because it's less efficient than not allowing a node type.

**Parameters**

| nodes | [in] A set of node identifiers that must be kept intact until the destruction of the class. |

3.11.2.9 - (id) initWithSession: (STSSession *) session node:(long long) node

Creates a new instance.

**Parameters**

| session | [in] Session to get the graph from and perform operation. |
| node | [in] Node to start traversal from. |

3.11.2.10 - (void) setMaximumHops: (int) maxhops include:(BOOL) include

Sets the maximum hops restriction.

All paths longer than the maximum hops restriction will be ignored.

**Parameters**

| maxhops | [in] The maximum hops restriction. It must be positive or zero. Zero, the default value, means unlimited. |
| include | [in] If TRUE, the resulting collection will include those nodes at distance less or equal than the given one, otherwise it will just include those nodes at distance equal than the given one. This parameter just makes sense if maxhops is different from 0; in that case it includes all nodes no matters the distance. |

The documentation for this class was generated from the following file:

- Sparksee.h

3.12 STSCSVReader Class Reference

CSVReader interface.
Inheritance diagram for STSCSVReader:

Collaboration diagram for STSCSVReader:

Instance Methods

- (id) - init
  Constructs CSVReader.
- (void) - setSeparator:
  Sets the character used to separate fields in the file.
- (void) - setQuotes:
  Sets the character used to quote fields.
- (void) - setMultilines:
  Allows the use of fields with more than one line.
- (void) - setSingleLine
Only allows single line fields.

- (void) - setStartLine:
  Sets the number of lines to be skipped from the beginning.
- (void) - setNumLines:
  Used to limit the number of lines that will be read.
- (void) - setLocale:
  Sets the locale that will be used to read the file.
- (void) - open:
  Opens the source file path.
- (BOOL) - reset
  Moves the reader to the beginning.
- (BOOL) - read:
  Reads the next row as a string array.
- (int) - getRow
  The row number for the current row.
- (void) - close
  Closes the reader.

3.12.1 Detailed Description

 CSVReader interface.

A very simple CSV reader.

It works as any other RowReader, but open must be called once before the first read operation.

Using the format RFC 4180.

Except: leading and trailing spaces, adjacent to CSV separator character, are trimmed.

You can use your own separators and quote characters. By default the separator is the comma (,) and the quote character is the double quotes (").

Fields with multiple lines can be allowed (and the maximum lines specified), but the default is a single line.

The locale string can be used to set the language, country and the file encoding. The format must be "[language-_territory][.codeset]". But only the file encoding is being used in the current version.

The languages supported are: "en_US", "es_ES" and "ca_ES".

The file encodings supported are: "utf8" and "iso88591".

For example:

To don't change the default locales, use an empty string: "".

To read a file in utf8 with the default language settings use ".utf8".

To read a file in iso88591 with English language use: "en_US.iso88591".

Check out the 'Data import' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies [http://www.sparsity-technologies.com](http://www.sparsity-technologies.com)

3.12.2 Method Documentation

3.12.2.1 - (void) close

Closes the reader.
3.12 STSCSVReader Class Reference

Exceptions

- `System.IO.IOException` If the close fails.

Implements `STSRowReader`.

3.12.2.2 - (int) `getRow`

The row number for the current row.

Returns

The current row number; 0 if there is no current row.

Exceptions

- `System.IO.IOException` If it fails.

Implements `STSRowReader`.

3.12.2.3 - (void) `open: (NSString *)filePath`

Opens the source file path.

File can be optionally compressed in GZIP format.

Parameters

- `filePath` [in] CSV file path.

Exceptions

- `System.IO.IOException` If bad things happen opening the file.

3.12.2.4 - (BOOL) `read: (STSStringList *)row`

Reads the next row as a string array.

Parameters

- `row` [out] A string list with each comma-separated element as a separate entry.

Returns

Returns true if a row had been read or false otherwise.

Exceptions

- `System.IO.IOException` If bad things happen during the read.

Implements `STSRowReader`.

3.12.2.5 - (BOOL) `reset`

Moves the reader to the beginning.

Restarts the reader.
Returns

true if the reader can be restarted, false otherwise.

Exceptions

- `System.IO.IOException` If bad things happen during the restart.

Implements `STSRowReader`.

3.12.2.6 - (void) setLocale: (NSString *) localeStr

Sets the locale that will be used to read the file.

Parameters

- `localeStr` [in] The locale string for the file encoding.

3.12.2.7 - (void) setMultilines: (int) numExtralines

Allows the use of fields with more than one line.

Parameters

- `numExtralines` [in] Maximum number of extra lines for each column (0==unlimited, N==N+1 total rows).

3.12.2.8 - (void) setNumLines: (int) numLines

Used to limit the number of lines that will be read.

Parameters

- `numLines` [in] The maximum number of lines to read (0 == unlimited)

3.12.2.9 - (void) setQuotes: (NSString *) quotes

Sets the character used to quote fields.

Parameters


Exceptions

- `System.ApplicationException` null

3.12.2.10 - (void) setSeparator: (NSString *) sep

Sets the character used to separate fields in the file.

Parameters


Exceptions

- `System.ApplicationException` null
3.12.2.11 - (void) setStartLine: (int) startLine

Sets the number of lines to be skiped from the beginning.

Parameters

| startLine | [in] The line number to skip for start reading |

The documentation for this class was generated from the following file:

- Sparksee.h

3.13 STSCSVWriter Class Reference

CSVWriter interface.

Inheritance diagram for STSCSVWriter:
3.13 STSCSVWriter Class Reference

Collaboration diagram for STSCSVWriter:

![Collaboration diagram](image)

**Instance Methods**

- (id) - init
  
  Creates a new instance.

- (void) - setSeparator:

  Sets the character used to separate fields in the file.

- (void) - setQuotes:

  Sets the character used to quote fields.

- (void) - setAutoQuotes:

  Sets on/off the automatic quote mode.

- (void) - setForcedQuotes:

  Disables the automatic quote mode and forces to be quoted those positions set to TRUE in the given vector.

- (void) - setLocale:

  Sets the locale that will be used to write the file.

- (void) - open:

  Opens the output file path.

- (void) - write:

  Writes the next row.

- (void) - close

  Closes the writer.

3.13.1 Detailed Description

CSVWriter interface.

A very simple CSV writer implementing RowWriter.

It works as any other RowWriter, but open must be called once before the first write operation.


You can use your own separators and quote characters. By default the separator is the comma (,) and the quote character is the double quotes (") and autoquote is enabled.
See the CSVReader locale documentation or the SPARKSEE User Manual.
Check out the 'Data export' section in the SPARKSEE User Manual for more details on this.

Author
Sparsity Technologies http://www.sparsity-technologies.com

3.13.2 Method Documentation

3.13.2.1 - (void) close
Closes the writer.

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.IO.IOException</td>
<td>If the close fails.</td>
</tr>
<tr>
<td>System.ApplicationException</td>
<td>null</td>
</tr>
</tbody>
</table>

Implements STSRowWriter.

3.13.2.2 - (void) open: (NSString *) f
Opens the output file path.

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>f</td>
<td>[in] Output file path.</td>
</tr>
</tbody>
</table>

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.IO.IOException</td>
<td>If bad things happen opening the file.</td>
</tr>
</tbody>
</table>

3.13.2.3 - (void) setAutoQuotes: (BOOL) autoquotes
Sets on/off the automatic quote mode.
If there are forced quotes, setting autoquotes on will clear them. If the autoquotes is set to off and no forced quotes are provided, there will not be any quote.

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>autoquotes</td>
<td>[in] If TRUE it enables the automatic quote mode, if FALSE it disables it.</td>
</tr>
</tbody>
</table>

3.13.2.4 - (void) setForcedQuotes: (STSSBooleanList *) forcequotes
Disables the automatic quote mode and forces to be quoted those positions set to TRUE in the given vector.

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>forcequotes</td>
<td>[in] A booleanList with the position for each column that must be quoted set to true.</td>
</tr>
</tbody>
</table>

3.13.2.5 - (void) setLocale: (NSString *) localeStr
Sets the locale that will be used to write the file.
Parameters

localeStr [in] The locale string for the file encoding.

3.13.2.6 - (void) setQuotes: (NSString *) quotes

Sets the character used to quote fields.

Parameters

quotes [in] Quote character.

Exceptions

System.Application- null
Exception

3.13.2.7 - (void) setSeparator: (NSString *) sep

Sets the character used to separate fields in the file.

Parameters

sep [in] Separator character.

Exceptions

System.Application- null
Exception

3.13.2.8 - (void) write: (STSStringList *) row

Writes the next row.

Parameters

row [in] Row of data.

Exceptions

System.IO.IOException If bad things happen during the write.
System.Application- null
Exception

Implements STSRowWriter.

The documentation for this class was generated from the following file:

• Sparksee.h

3.14 STSDatabase Class Reference

Database class.
Inheritance diagram for STSDatabase:

```
Inheritance diagram for STSDatabase:

NSObject

STSDatabase
```

Collaboration diagram for STSDatabase:

```
Collaboration diagram for STSDatabase:

NSObject

STSDatabase
```

Instance Methods

- `(NSString *) - getAlias`
  
  *Gets the alias of the Database.*

- `(NSString *) - getPath`
  
  *Gets the path of the Database.*

- `(STSSession *) - createSession`
  
  *Creates a new Session.*

- `(void) - enableRollback`
  
  *Enables the rollback mechanism.*

- `(void) - disableRollback`
  
  *Disables the rollback mechanism.*

- `(void) - getStatistics:
  
  *Gets Database statistics.*

- `(int) - getCacheMaxSize`
  
  *Gets the cache maximum size (in MB).*

- `(void) - setCacheMaxSize:
  
  *Sets the cache maximum size (in MB).*

- `(void) - fixCurrentCacheMaxSize`
Sets the cache maximum size to the current cache size in use.

- (void) close
  Closes the Database instance.

- (BOOL) isClosed
  Check if the Database instance is closed.

3.14.1 Detailed Description

Database class.
All the data of the Database is stored into a persistent file which just can be created or open through a Sparksee instance.
Also, all the manipulation of a Database must be done by means of a Session which can be initiated from a Database instance.
Multiple Databases do not share the memory, that is there is no negotiation among them. In those cases, memory must be prefixed for each Database. To do that, use the SPARKSEEConfig.

Author
Sparsity Technologies [http://www.sparsity-technologies.com](http://www.sparsity-technologies.com)

3.14.2 Method Documentation

3.14.2.1 - (void) close
Closes the Database instance.
It must be called to ensure the integrity of all data.

3.14.2.2 - (void) fixCurrentCacheMaxSize
Sets the cache maximum size to the current cache size in use.

Returns
  Returns true if successful or false otherwise.

3.14.2.3 - (NSString*) getAlias
Gets the alias of the Database.

Returns
  The alias of the Database.

3.14.2.4 - (int) getCacheMaxSize
Gets the cache maximum size (in MB).

Returns
  Returns the current cache max size.

3.14.2.5 - (NSString*) getPath
Gets the path of the Database.

Returns
  The path of the Database.
3.14.2.6 - (void) getStatistics: (STSDatabaseStatistics *) stats

Gets Database statistics.

Parameters


3.14.2.7 - (void) setCacheMaxSize: (int) megaBytes

Sets the cache maximum size (in MB).
0 means unlimited which is all the physical memory of the computer minus a small margin.

Parameters


The documentation for this class was generated from the following file:

- Sparksee.h

3.15 STSDatabaseStatistics Class Reference

Database statistics.

Inheritance diagram for STSDatabaseStatistics:
Collaboration diagram for STSDatabaseStatistics:

```
ysqli
```

**Instance Methods**

- `(long long) - getRead`  
  *Gets total read data in KBytes.*

- `(long long) - getWrite`  
  *Gets total written data in KBytes.*

- `(long long) - getData`  
  *Gets database size in KBytes.*

- `(long long) - getCache`  
  *Gets cache size in KBytes.*

- `(long long) - getTemp`  
  *Gets temporary storage file size in KBytes.*

- `(long long) - getSessions`  
  *Gets the number of sessions.*

### 3.15.1 Detailed Description

Database statistics.

**Author**

Sparsity Technologies [http://www.sparsity-technologies.com](http://www.sparsity-technologies.com)

### 3.15.2 Method Documentation

#### 3.15.2.1 `(long long) getCache`

Gets cache size in KBytes.

**Returns**

Cache size in KBytes.
3.15.2.2 - (long long) getData

Gets database size in KBytes.

Returns

Database size in KBytes.

3.15.2.3 - (long long) getRead

Gets total read data in KBytes.

Returns

Total read data in KBytes.

3.15.2.4 - (long long) getSessions

Gets the number of sessions.

Returns

The number of sessions.

3.15.2.5 - (long long) getTemp

Gets temporary storage file size in KBytes.

Returns

Temporary storage file size in KBytes.

3.15.2.6 - (long long) getWrite

Gets total written data in KBytes.

Returns

Total read written in KBytes.

The documentation for this class was generated from the following file:

• Sparksee.h

3.16 STSDefaultExport Class Reference

Default implementation for ExportManager class.
Inheritance diagram for STSDefaultExport:

```
NSObject

STSExportManager

STSDefaultExport
```

Collaboration diagram for STSDefaultExport:

```
NSObject

STSExportManager

STSDefaultExport
```

Instance Methods

- (id) init
  
  Creates a new instance.

- (void) prepare:
  
  Default implementation of the ExportManager class method Prepare.

- (void) close
  
  Default implementation of the ExportManager class method Release.

- (BOOL) getGraph:
  
  Default implementation of the ExportManager class method GetGraph.

- (BOOL) getType:nodeExport:
3.16 STSDefaultExport Class Reference

Default implementation of the ExportManager class method GetNodeType.

• (BOOL) - getNodeType:edgeExport:
  Default implementation of the ExportManager class method GetNodeType.

• (BOOL) - getNode:nodeExport:
  Default implementation of the ExportManager class method GetNode.

• (BOOL) - getEdge:edgeExport:
  Default implementation of the ExportManager class method GetEdge.

• (BOOL) - enableType:
  Default implementation of the ExportManager class method EnableType.

3.16.1 Detailed Description

Default implementation for ExportManager class.

It uses the default values from GraphExport, NodeExport and EdgeExport to export all node and edge types.

Author

Sparsity Technologies http://www.sparsity-technologies.com

3.16.2 Method Documentation

3.16.2.1 - (BOOL) enableType: (int) type

Default implementation of the ExportManager class method EnableType.

This enables all node and edge types to be exported.

Parameters

| type | [in] The type to enable. |

Returns

TRUE.

Implements STSExportManager.

3.16.2.2 - (BOOL) getEdge: (long long) edge: (STSEdgeExport *) edgeExport

Default implementation of the ExportManager class method GetEdge.

This sets the default EdgeExport values and sets the OID as the label. Also, it exports the edge as directed just if
the edge is directed.

Parameters

| edgeExport | [out] The EdgeExport that will store the information. |

Returns

TRUE.

Implements STSExportManager.

3.16.2.3 - (BOOL) getType: (int) typeExport:(STSEdgeExport *) edgeExport

Default implementation of the ExportManager class method GetEdgeType.
This sets the default EdgeExport values.

Parameters

<table>
<thead>
<tr>
<th>type</th>
<th>[in] An edge type.</th>
</tr>
</thead>
<tbody>
<tr>
<td>edgeExport</td>
<td>[out] The EdgeExport that will store the information.</td>
</tr>
</tbody>
</table>

Returns

TRUE.

Implements STSExportManager.

3.16.2.4 - (BOOL) getGraph: (STSGraphExport *) graphExport

Default implementation of the ExportManager class method GetGraph.
This sets the default GraphExport values and “Graph” as the label.

Parameters

| graphExport | [out] The GraphExport that will store the information. |

Returns

TRUE.

Implements STSExportManager.

3.16.2.5 - (BOOL) getNode: (long long) node nodeExport:(STSNodeExport *) nodeExport

Default implementation of the ExportManager class method GetNode.
This sets the default NodeExport values and sets the OID as the label.

Parameters

| node | [in] A node. |
| nodeExport | [out] The NodeExport that will store the information. |

Returns

TRUE.

Implements STSExportManager.

3.16.2.6 - (BOOL) getNodeType: (int) type nodeExport:(STSNodeExport *) nodeExport

Default implementation of the ExportManager class method GetNodeType.
This sets the default NodeExport values.

Parameters

| type | [in] A node type. |
| nodeExport | [out] The NodeExport that will store the information. |
Returns
TRUE.

Implements STSExportManager.

3.16.2.7 - (void) prepare: (STSGraph *) graph
Default implementation of the ExportManager class method Prepare.

Parameters

| graph | null |

Implements STSExportManager.
The documentation for this class was generated from the following file:

- Sparksee.h

3.17 STSDisjointCommunities Class Reference

DisjointCommunities class.
Inheritance diagram for STSDisjointCommunities:

Collaboration diagram for STSDisjointCommunities:
3.17 STDisjointCommunities Class Reference

Instance Methods

- (id) initWithSession:materializedattribute:
  Creates a new instance of DisjointCommunities.

- (long long) getCommunity:
  Returns the disjoint community where the given node belongs to.

- (long long) getCount
  Returns the number of communities found in the graph.

- (STSObjects *) getNodes:
  Returns the collection of nodes contained in the given community.

- (long long) getSize:
  Returns the number of nodes contained in the given community.

- (void) close
  Closes the DisjointCommunities instance.

- (BOOL) isClosed
  Check if the DisjointCommunities instance is closed.

3.17.1 Detailed Description

DisjointCommunities class.

This class contains the results processed on a DisjointCommunityDetection algorithm.

These results contain information related to the communities found. We must consider that each community has a
number in order to identify it. These number identifiers are values from 0 to N-1, where N is the number of different
communities found.

When executing any implementation of the DisjointCommunityDetection, it is possible to indicate whether the results
of the execution must be stored persistently using the class DisjointCommunityDetection setMaterializedAttribute
method. In case the results are set to be materialized, users can retrieve this data whenever they want, even if the
graph has been closed and opened again, just by creating a new instance of this class.

Check out the ’Algorithms’ section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com

3.17.2 Method Documentation

3.17.2.1 - (void) close

Closes the DisjointCommunities instance.

It must be called to ensure the integrity of all data.

3.17.2.2 - (long long) getCommunity: (long long) idNode

Returns the disjoint community where the given node belongs to.

Parameters

| idNode   | [in] The node identifier for which the disjoint community identifier where it belongs will be returned. |

Returns

The disjoint community identifier where the given node identifier belongs to.
### 3.17.2.3 - (long long) getCount

Returns the number of communities found in the graph.

**Returns**

The number of communities found in the graph.

### 3.17.2.4 - (STSObjects*) getNodes: (long long) idCommunity

Returns the collection of nodes contained in the given community.

**Parameters**

| idCommunity | The community for which the collection of nodes contained in it will be returned. |

**Returns**

The collection of node identifiers contained in the given community.

### 3.17.2.5 - (long long) getSize: (long long) idCommunity

Returns the number of nodes contained in the given community.

**Parameters**

| idCommunity | The community for which the number of nodes contained in it will be returned. |

**Returns**

The number of nodes contained in the given community.

### 3.17.2.6 - (id) initWithSession: (STSSession*) session materializedattribute:(NSString*) materializedattribute

Creates a new instance of DisjointCommunities.

This constructor method can only be called when a previous execution of any implementation of the DisjointCommunityDetection class has materialized the results in a common attribute type for all the nodes in the graph. For further information about materializing the results processed on any DisjointCommunityDetection execution see the documentation of the DisjointCommunityDetection::SetMaterializedAttribute method.

**Parameters**

| session | [in] Session to get the graph Graph on which the information will be retrieved just by getting the values contained in the given common attribute type for all the nodes in the graph and processing them. |
| materializedattribute | [in] The common attribute type for all the nodes in the graph where data will be retrieved in order to process the results related to the communities found in the graph. |

The documentation for this class was generated from the following file:

- Sparksee.h

### 3.18 STSDisjointCommunityDetection Class Reference

DisjointCommunityDetection class.
Inheritance diagram for STSDisjointCommunityDetection:

Collaboration diagram for STSDisjointCommunityDetection:

Instance Methods

- (void) - addEdgeType:
  Allows connectivity through edges of the given type.
- (void) - addAllEdgeTypes
  Allows connectivity through all edge types of the graph.
• (STSDisjointCommunities +) - getCommunities
  Returns the results generated by the execution of the algorithm.
• (void) - run
  Runs the algorithm in order to find the communities.
• (void) - setMaterializedAttribute:
  Creates a new common attribute type for all node types in the graph in order to store, persistently, the results related to the disjoint communities found while executing this algorithm.
• (void) - addNodeType:
  Allows connectivity through nodes of the given type.
• (void) - addAllNodeTypes
  Allows connectivity through all node types of the graph.
• (void) - excludeNodes:
  Set which nodes can’t be used.
• (void) - excludeEdges:
  Set which edges can’t be used.
• (void) - close
  Closes the CommunityDetection instance.
• (BOOL) - isClosed
  Check if the CommunityDetection instance is closed.

3.18.1 Detailed Description

DisjointCommunityDetection class.

Any class implementing this abstract class can be used to solve a problem related to graph connectivity as finding the strongly connected components, finding the weakly connected components.

Check out the ‘Algorithms’ section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com

3.18.2 Method Documentation

3.18.2.1 - (void) addAllEdgeTypes

Allows connectivity through all edge types of the graph.
The edges can be used in Any direction.

3.18.2.2 - (void) addEdgeType: (int) type

Allows connectivity through edges of the given type.
The edges can be used in Any direction.

Parameters

| type | [in] Edge type. |

3.18.2.3 - (void) addNodeType: (int) type

Allows connectivity through nodes of the given type.
3.18.2.4 - (void) close

Closes the CommunityDetection instance. It must be called to ensure the integrity of all data.

3.18.2.5 - (void) excludeEdges: (STSObjects *) edges

Set which edges can’t be used. This will replace any previously specified set of excluded edges. Should only be used to exclude the usage of specific edges from allowed edge types because it’s less efficient than not allowing an edge type.

Parameters

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>edges</td>
<td>[in] A set of edge identifiers that must be kept intact until the destruction of the class.</td>
</tr>
</tbody>
</table>

3.18.2.6 - (void) excludeNodes: (STSObjects *) nodes

Set which nodes can’t be used. This will replace any previously specified set of excluded nodes. Should only be used to exclude the usage of specific nodes from allowed node types because it’s less efficient than not allowing a node type.

Parameters

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>nodes</td>
<td>[in] A set of node identifiers that must be kept intact until the destruction of the class.</td>
</tr>
</tbody>
</table>

3.18.2.7 - (STSDisjointCommunities *) getCommunities

Returns the results generated by the execution of the algorithm. These results contain information related to the disjoint communities found as the number of different components, the set of nodes contained in each component or many other data.

Returns

Returns an instance of the class DisjointCommunities which contain information related to the disjoint communities found.

3.18.2.8 - (void) run

Runs the algorithm in order to find the communities. This method can be called only once.

Implements STSCommunityDetection.

Implemented in STSCommunitiesSCD.

3.18.2.9 - (void) setMaterializedAttribute: (NSString *) attributeName

Creates a new common attribute type for all node types in the graph in order to store, persistently, the results related to the disjoint communities found while executing this algorithm.

Whenever the user wants to retrieve the results, even when the graph has been closed and opened again, it is only necessary to create a new instance of the class DisjointCommunities indicating the graph and the name of the common attribute type which stores the results. This instance will have all the information related to the disjoint communities found in the moment of the execution of the algorithm that stored this data.
It is possible to run the algorithm without specifying this parameter in order to avoid materializing the results of the execution.

Parameters

| attributeName | [in] The name of the common attribute type for all node types in the graph which will store persistently the results generated by the execution of the algorithm. |

The documentation for this class was generated from the following file:

- Sparksee.h

3.19 STSEdgeData Class Reference

Edge data class.

Inheritance diagram for STSEdgeData:

```
NSObject

STSEdgeData
```

Collaboration diagram for STSEdgeData:

```
NSObject

STSEdgeData
```

Instance Methods

- (long long) - **getEdge**
  
  *Gets the edge identifier.*
- (long long) - **getTail**
3.20 STSEdgeExport Class Reference

Stores edge exporting values.

3.19.1 Detailed Description

Edge data class.
It stores the tail and the head of an edge instance.
In case of undirected edges, the tail and the head are just the two ends of the edge.

Author
Sparsity Technologies http://www.sparsity-technologies.com

3.19.2 Method Documentation

3.19.2.1 - (long long) getEdge

Gets the edge identifier.

Returns
The Sparksee edge identifier.

3.19.2.2 - (long long) getHead

Gets the head of the edge.

Returns
The Sparksee edge identifier of the head of the edge.

3.19.2.3 - (long long) getTail

Gets the tail of the edge.

Returns
The Sparksee edge identifier of the tail of the edge.

The documentation for this class was generated from the following file:

• Sparksee.h
Inheritance diagram for STSEdgeExport:

```
NSObject
  ↓
STSEdgeExport
```

Collaboration diagram for STSEdgeExport:

```
NSObject
  ↓
STSEdgeExport
```

Instance Methods

- (id) - init
  Creates a new instance.
- (void) - setDefaults
  Sets to default values.
- (NSString *) - getLabel
  Gets the edge label.
- (void) - setLabel:
  Sets the edge label.
- (BOOL) - asDirected
  Gets if the edge should be managed as directed.
- (void) - setAsDirected:
  Sets if the edge should be managed as directed.
- (int) - getColorRGB
  Gets the edge color.
- (void) - setColorRGB:
  Sets the edge color.
- (int) - getLabelColorRGB
3.20 STSEdgeExport Class Reference

3.20.1 Detailed Description

Stores edge exporting values.

Some properties may be ignored depending on the exportation type.

Default values are:

- Label: "" (empty string).
- As directed: TRUE.
- Color: 13882323 (0xD3D3D3, Light gray).
- Label color: 0 (0x000000, Black).
- Width: 5px.
- Font size: 10.

Author

Sparsity Technologies [http://www.sparsity-technologies.com](http://www.sparsity-technologies.com)

3.20.2 Method Documentation

3.20.2.1 -(BOOL) asDirected

Gets if the edge should be managed as directed.

TRUE is the default value. If TRUE, use as directed, otherwise use as undirected.

Returns

The edge direction.
3.20.2.2 - (void) getColor: (double *) red green:(double *) green blue:(double *) blue alpha:(double *) alpha

Get the edge color separated in RGBA.

Parameters

<table>
<thead>
<tr>
<th>red</th>
<th>[out] The red color component ([0..1]).</th>
</tr>
</thead>
<tbody>
<tr>
<td>green</td>
<td>[out] The green color component ([0..1]).</td>
</tr>
<tr>
<td>blue</td>
<td>[out] The blue color component ([0..1]).</td>
</tr>
<tr>
<td>alpha</td>
<td>[out] The alpha component ([0..1]).</td>
</tr>
</tbody>
</table>

3.20.2.3 - (int) getColorRGB

Gets the edge color.

Returns

The edge color.

3.20.2.4 - (int) getFontSize

Gets the edge label font size.

Returns

The edge label font size.

3.20.2.5 - (NSString *) getLabel

Gets the edge label.

Returns

The edge label.

3.20.2.6 - (void) getLabelColor: (double *) red green:(double *) green blue:(double *) blue alpha:(double *) alpha

Get the edge label color separated in RGBA.

Parameters

<table>
<thead>
<tr>
<th>red</th>
<th>[out] The red color component ([0..1]).</th>
</tr>
</thead>
<tbody>
<tr>
<td>green</td>
<td>[out] The green color component ([0..1]).</td>
</tr>
<tr>
<td>blue</td>
<td>[out] The blue color component ([0..1]).</td>
</tr>
<tr>
<td>alpha</td>
<td>[out] The alpha component ([0..1]).</td>
</tr>
</tbody>
</table>

3.20.2.7 - (int) getLabelColorRGB

Gets the edge label color.

Returns

The edge label color.

3.20.2.8 - (int) getWidth

Gets the edge width.
3.20.2.9 - (void) setAsDirected: (BOOL) directed

Sets if the edge should be managed as directed.

Parameters

| directed | [in] If TRUE, use as directed, otherwise use as undirected. |

3.20.2.10 - (void) setColorRed: (double) red green:(double) green blue:(double) blue alpha:(double) alpha

Set the edge color with separated RGBA components.

Parameters

| red       | [in] The red color component ([0..1]). |
| green     | [in] The green color component ([0..1]). |
| blue      | [in] The blue color component ([0..1]). |
| alpha     | [in] The alpha component ([0..1]). |

3.20.2.11 - (void) setColorRGB: (int) color

Sets the edge color.

Parameters

| color     | [in] The edge color. |

3.20.2.12 - (void) setFontSize: (int) size

Sets the edge label font size.

Parameters

| size      | [in] The edge label font size. |

3.20.2.13 - (void) setLabel: (NSString *) label

Sets the edge label.

Parameters

| label     | [in] The edge label. |

3.20.2.14 - (void) setLabelColorRed: (double) red green:(double) green blue:(double) blue alpha:(double) alpha

Set the edge label color with separated RGBA components.

Parameters

| red       | [in] The red color component ([0..1]). |
| green     | [in] The green color component ([0..1]). |
| blue      | [in] The blue color component ([0..1]). |
| alpha     | [in] The alpha component ([0..1]). |
3.20.2.15 - (void) setLabelColorRGB: (int) color

Sets the edge label color.

Parameters

| color | [in] The edge label color. |

3.20.2.16 - (void) setWidth: (int) width

Sets the edge width.

Parameters

| width | [in] The edge width. |

The documentation for this class was generated from the following file:

- Sparksee.h

3.21 STSEdgeTypeExporter Class Reference

EdgeTypeExporter class.

Inheritance diagram for STSEdgeTypeExporter:
Collaboration diagram for STSEdgeTypeExporter:

- NSObject
  - STSTypeExporter
    - STSEdgeTypeExporter

Instance Methods

- (id) - init
  Creates a new instance.

- (id) - initWithRowWriter:graph:type:attrs:hPos:tPos:hAttr:tAttr:
  Creates a new instance.

- (void) - run
  See the TypeExporter class Run method.

- (void) - setHeadAttribute:
  Sets the attribute that will be used to get the value to be dumped for the head of the edge.

- (void) - setHeadPosition:
  Sets the position (index column) of the head attribute in the exported data.

- (void) - setTailAttribute:
  Sets the attribute that will be used to get the value to be dumped for the tail of the edge.

- (void) - setTailPosition:
  Sets the position (index column) of the tail attribute in the exported data.

- (void) - registerListener:
  Registers a new listener.

- (void) - setRowWriter:
  Sets the output data destination.

- (void) - setGraph:
  Sets the graph that will be exported.

- (void) - setType:
  Sets the type to be exported.

- (void) - setAttributes:
  Sets the list of Attributes.

- (void) - setFrequency:
  Sets the frequency of listener notification.

- (void) - setHeader:
  Sets the presence of a header row.
3.21 STSEdgeTypeExporter Class Reference

3.21.1 Detailed Description

EdgeTypeExporter class.
Specific TypeExporter implementation for edge types.
Check out the 'Data export' section in the SPARKSEE User Manual for more details on this.

Author
Sparsity Technologies http://www.sparsity-technologies.com

3.21.2 Method Documentation

3.21.2.1 - (id) initWithRowWriter: (STSRowWriter *) rowWriter graph:(STSGraph *) graph type:(int) type attrs:(STSAttributeList *) attrs hPos:(int) hPos tPos:(int) tPos hAttr:(int) hAttr tAttr:(int) tAttr

Creates a new instance.

Parameters

| graph     | [in] Graph. |
| type      | [in] Type identifier. |
| attrs     | [in] Attribute identifiers to be exported. |
| hPos      | [in] The position (index column) for the head value. |
| tPos      | [in] The position (index column) for the tail value. |
| hAttr     | [in] The attribute identifier to get the value to be dumped for the head. |
| tAttr     | [in] The attribute identifier to get the value to be dumped for the tail. |

3.21.2.2 - (void) registerListener: (STSTypeExporterListener *) tel

Registers a new listener.

Parameters

| tel | [in] TypeExporterListener to be registered. |

3.21.2.3 - (void) run

See the TypeExporter class Run method.

Exceptions

| System.IO.IOException | null |
| System.ApplicationException | null |

Implements STSTypeExporter.

3.21.2.4 - (void) setAttributes: (STSAttributeList *) attrs

Sets the list of Attributes.

Parameters

| attrs | [in] Attribute identifiers to be exported |
3.21.2.5 - (void) setFrequency: (int) freq
Sets the frequency of listener notification.

Parameters

| freq | [in] Frequency in number of rows managed to notify progress to all listeners |

3.21.2.6 - (void) setGraph: (STSGraph *) graph
Sets the graph that will be exported.

Parameters

| graph | [in] Graph. |

3.21.2.7 - (void) setHeadAttribute: (int) attr
Sets the attribute that will be used to get the value to be dumped for the head of the edge.

Parameters

| attr | [in] Head Attribute |

3.21.2.8 - (void) setHeader: (BOOL) header
Sets the presence of a header row.

Parameters

| header | [in] If TRUE, a header row is dumped with the name of the attributes. |

3.21.2.9 - (void) setHeadPosition: (int) pos
Sets the position (index column) of the head attribute in the exported data.

Parameters

| pos | [in] Head position |

3.21.2.10 - (void) setRowWriter: (STSRowWriter *) rw
Sets the output data destination.

Parameters

| rw | [in] Input RowWriter. |

3.21.2.11 - (void) setTailAttribute: (int) attr
Sets the attribute that will be used to get the value to be dumped for the tail of the edge.

Parameters

| attr | [in] Tail Attribute |
3.21.12 - (void) setTailPosition: (int) pos

Sets the position (index column) of the tail attribute in the exported data.

Parameters

| pos | [in] Tail position |

3.21.13 - (void) setType: (int) type

Sets the type to be exported.

Parameters

| type | [in] Type identifier. |

The documentation for this class was generated from the following file:

- Sparksee.h

3.22 STSEdgeTypeLoader Class Reference

EdgeTypeLoader class.

Inheritance diagram for STSEdgeTypeLoader:
Collaboration diagram for STSEdgeTypeLoader:

Instance Methods

- (id) - init
  Creates a new instance.

- (id) - initWithRowReader:graph:type:attrs:attrsPos:hPos:hPos:hAttr:tAttr:
  Creates a new instance.

- (void) - run
  See the TypeLoader class Run method.

- (void) - runTwoPhases
  See the TypeLoader class RunTwoPhases method.

- (void) - runNPhases:
  See the TypeLoader class RunNPhases method.

- (void) - setHeadAttribute:
  Sets the attribute that will be used to find the head of the edge.

- (void) - setHeadPosition:
  Sets the position of the head attribute in the source data.

- (void) - setTailAttribute:
  Sets the attribute that will be used to find the tail of the edge.

- (void) - setTailPosition:
  Sets the position of the tail attribute in the source data.

- (void) - setLogError:
  Sets a log error file.

- (void) - setLogOff
  Turns off all the error reporting.

- (void) - registerListener:
  Registers a new listener.

- (void) - setRowReader:
  Sets the input data source.

- (void) - setGraph:
  Sets the graph where the data will be loaded.

- (void) - setLocale:
Sets the locale that will be used to read the data.

- (void) - setType:
  Sets the type to be loaded.

- (void) - setAttributes:
  Sets the list of Attributes.

- (void) - setAttributePositions:
  Sets the list of attribute positions.

- (void) - setTimestampFormat:
  Sets a specific timestamp format.

- (void) - setFrequency:
  Sets the frequency of listener notification.

### 3.22.1 Detailed Description

**EdgeTypeLoader class.**

Specific TypeLoader implementation for edge types.

Check out the ‘Data import’ section in the SPARKSEE User Manual for more details on this.

**Author**

Sparsity Technologies [http://www.sparsity-technologies.com](http://www.sparsity-technologies.com)

### 3.22.2 Method Documentation

#### 3.22.2.1 - (id) initWithRowReader:  
STSRowReader ∗ rowReader  
STSGraph ∗ graph:  
(int) type  
STSAttributeList ∗ attrs:  
STSInt32List ∗ attrsPos:  
(int) hPos  
(int) tPos  
(int) hAttr  
(int) tAttr

Creates a new instance.

**Parameters**

<table>
<thead>
<tr>
<th>rowReader</th>
<th>[in] Input RowReader.</th>
</tr>
</thead>
<tbody>
<tr>
<td>graph</td>
<td>[in] Graph.</td>
</tr>
<tr>
<td>type</td>
<td>[in] Type identifier.</td>
</tr>
<tr>
<td>attrs</td>
<td>[in] Attribute identifiers to be loaded.</td>
</tr>
<tr>
<td>attrsPos</td>
<td>[in] Attribute positions (column index &gt;=0). to all listeners.</td>
</tr>
<tr>
<td>hPos</td>
<td>[in] The position (index column) for the head value.</td>
</tr>
<tr>
<td>tPos</td>
<td>[in] The position (index column) for the tail value.</td>
</tr>
<tr>
<td>hAttr</td>
<td>[in] The attribute identifier for the head.</td>
</tr>
<tr>
<td>tAttr</td>
<td>[in] The attribute identifier for the tail.</td>
</tr>
</tbody>
</table>

#### 3.22.2.2 - (void) registerListener:  
STSTypeLoaderListener ∗ tel

Registers a new listener.

**Parameters**

| tel | TypeLoaderListener to be registered. |

#### 3.22.2.3 - (void) run

See the TypeLoader class Run method.
### Exceptions

- `System.IO.IOException`  
- `System.ApplicationException`  

Implements `STSTypeLoader`.

#### 3.22.2.4 - (void) runNPhases: (int) partitions

See the `TypeLoader` class `RunNPhases` method.

**Parameters**

- `partitions`  

**Exceptions**

- `System.IO.IOException`  
- `System.ApplicationException`  

Implements `STSTypeLoader`.

#### 3.22.2.5 - (void) runTwoPhases

See the `TypeLoader` class `RunTwoPhases` method.

**Exceptions**

- `System.IO.IOException`  
- `System.ApplicationException`  

Implements `STSTypeLoader`.

#### 3.22.2.6 - (void) setAttributePositions: (STSInt32List *) attrsPos

Sets the list of attribute positions.

**Parameters**

- `attrsPos`  

| attrsPos | [in] Attribute positions (column index >= 0). |

#### 3.22.2.7 - (void) setAttributes: (STSAtributeList *) attrs

Sets the list of Attributes.

**Parameters**

- `attrs`  

| attrs | [in] Attribute identifiers to be loaded |

#### 3.22.2.8 - (void) setFrequency: (int) freq

Sets the frequency of listener notification.

**Parameters**

- `freq`  

| freq | [in] Frequency in number of rows managed to notify progress to all listeners |
3.22.9 - (void) setGraph: (STSGraph *) graph

Sets the graph where the data will be loaded.

<table>
<thead>
<tr>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>graph</td>
</tr>
</tbody>
</table>

3.22.10 - (void) setHeadAttribute: (int) attr

Sets the attribute that will be used to find the head of the edge.
This method is protected because only the Edge loaders should have it.

<table>
<thead>
<tr>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>attr</td>
</tr>
</tbody>
</table>

3.22.11 - (void) setHeadPosition: (int) pos

Sets the position of the head attribute in the source data.
This method is protected because only the Edge loaders should have it.

<table>
<thead>
<tr>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>pos</td>
</tr>
</tbody>
</table>

3.22.12 - (void) setLocale: (NSString *) localeStr

Sets the locale that will be used to read the data.
It should match the locale used in the rowreader.

<table>
<thead>
<tr>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>localeStr</td>
</tr>
</tbody>
</table>

3.22.13 - (void) setLogError: (NSString *) path

Sets a log error file.
By default errors are thrown as an exception and the load process ends. If a log file is set, errors are logged there and the load process does not stop.

<table>
<thead>
<tr>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>path</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Exceptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.IO.IOException</td>
</tr>
</tbody>
</table>

3.22.14 - (void) setLogOff

Turns off all the error reporting.
The log file will not be created and no exceptions for invalid data will be thrown. If you just want to turn off the logs, but abort at the first error what you should do is not call this method and not set a logError file.
3.22.15 - (void) setRowReader: (STSRowReader *) rr

Sets the input data source.

Parameters

| rr | [in] Input RowReader. |

3.22.16 - (void) setTailAttribute: (int) attr

Sets the attribute that will be used to find the tail of the edge.
This method is protected because only the Edge loaders should have it.

Parameters

| attr | [in] Tail Attribute |

3.22.17 - (void) setTailPosition: (int) pos

Sets the position of the tail attribute in the source data.
This method is protected because only the Edge loaders should have it.

Parameters

| pos | [in] Tail position |

3.22.18 - (void) setTimestampFormat: (NSString *) timestampFormat

Sets a specific timestamp format.

Parameters

| timestampFormat | [in] A string with the timestamp format definition. |

3.22.19 - (void) setType: (int) type

Sets the type to be loaded.

Parameters

| type | [in] Type identifier. |

The documentation for this class was generated from the following file:

- Sparksee.h

3.23 STSExportManager Class Reference

Defines how to export a graph to an external format.
Inheritance diagram for STSExportManager:

```
NSObject

STSExportManager

STSEdefaultExport
```

Collaboration diagram for STSExportManager:

```
NSObject

STSExportManager
```

Instance Methods

- (void) - prepare:
  Prepares the graph for the export process.
- (void) - close
  Ends the export process.
- (BOOL) - getGraph:
  Gets the graph export definition.
- (BOOL) - getNodeType:nodeExport:
  Gets the default node export definition for the given node type.
- (BOOL) - getEdgeType:edgeExport:
  Gets the default node export definition for the given edge type.
- (BOOL) - getNode:nodeExport:
  Gets the node export definition for the given node.
- (BOOL) - getEdge:edgeExport:
3.23 STSExportManager Class Reference

Gets the edge export definition for the given edge.

- *(BOOL)* - enableType:
  
  Gets whether a node or edge type must be exported or not.

3.23.1 Detailed Description

Defines how to export a graph to an external format.

This is an interface which must be implemented by the user. While the export process, a call for each node or edge type and node or edge object is done to get how to export that element.

It is possible to export a Graph to a different formats. Nowadays, available formats are defined in the ExportType enum.

Author

Sparsity Technologies [http://www.sparsity-technologies.com](http://www.sparsity-technologies.com)

3.23.2 Method Documentation

3.23.2.1 - *(void)* close

Ends the export process.

It is called once after the export process.

Implemented in STSDefaultExport.

3.23.2.2 - *(BOOL)* enableType: *(int)* type

Gets whether a node or edge type must be exported or not.

Parameters

<table>
<thead>
<tr>
<th>type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node or edge type identifier.</td>
</tr>
</tbody>
</table>

Returns

If TRUE all objects of the given type will be exported, otherwise they will not be exported.

Implemented in STSDefaultExport.

3.23.2.3 - *(BOOL)* getEdge: *(long long)* edge edgeExport: *(STSEdgeExport *) edgeExport

Gets the edge export definition for the given edge.

Parameters

<table>
<thead>
<tr>
<th>edge</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edge identifier.</td>
</tr>
</tbody>
</table>

| edgeExport |
| [out] The EdgeExport which defines how to export given edge. |

Returns

TRUE if the given EdgeExport has been updated, otherwise FALSE will be returned and the default EdgeExport for the type the edge belongs to will be used.

Implemented in STSDefaultExport.

3.23.2.4 - *(BOOL)* getType: *(int)* type edgeExport: *(STSEdgeExport *) edgeExport

Gets the default node export definition for the given edge type.
GetEdge has a higher priority than this function. That is, only if GetEdge returns FALSE, the EdgeExport of this function will be used.

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>[in] Edge type identifier.</td>
</tr>
<tr>
<td>edgeExport</td>
<td>[out] The EdgeExport which defines how to export the edges of the given type.</td>
</tr>
</tbody>
</table>

Returns

TRUE.

Implemented in STSDefaultExport.

3.23.2.5 - (BOOL) getGraph: (STSGraphExport ∗) graphExport

Gets the graph export definition.

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>graphExport</td>
<td>[out] The GraphExport which defines how to export the graph.</td>
</tr>
</tbody>
</table>

Returns

TRUE.

Implemented in STSDefaultExport.

3.23.2.6 - (BOOL) getNode: (long long) node nodeExport:(STSNodeExport ∗) nodeExport

Gets the node export definition for the given node.

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>node</td>
<td>Node identifier.</td>
</tr>
<tr>
<td>nodeExport</td>
<td>[out] The NodeExport which defines how to export given node.</td>
</tr>
</tbody>
</table>

Returns

TRUE if the given NodeExport has been updated, otherwise FALSE will be returned and the default NodeExport for the type the node belongs to will be used.

Implemented in STSDefaultExport.

3.23.2.7 - (BOOL) getNodeType: (int) type nodeExport:(STSNodeExport ∗) nodeExport

Gets the default node export definition for the given node type.

GetNode has a higher priority than this function. That is, only if GetNode returns FALSE, the NodeExport of this function will be used.

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>[in] Node type identifier.</td>
</tr>
<tr>
<td>nodeExport</td>
<td>[out] The NodeExport which defines how to export the nodes of the given type.</td>
</tr>
</tbody>
</table>
Returns

TRUE.

Implemented in STSDefaultExport.

3.23.2.8 - (void) prepare: (STSGraph *) graph

Prepares the graph for the export process.
It is called once before the export process.

Parameters

| graph | Graph to be exported. |

Implemented in STSDefaultExport.

The documentation for this class was generated from the following file:

- Sparksee.h

3.24 STSGraph Class Reference

Graph class.

Inheritance diagram for STSGraph:

Collaboration diagram for STSGraph:
Instance Methods

- (int) - `createNodeType`:
  Creates a new node type.

- (long long) - `createNode`:
  Creates a new node instance.

- (int) - `createEdgeType:directed:neighbors`:
  Creates a new edge type.

- (int) - `createRestrictedEdgeType:tail:head:neighbors`:
  Creates a new restricted edge type.

- (long long) - `createEdge:tail:head`:
  Creates a new edge instance.

- (long long) - `createEdgeWithAttributes:tailAttr:tailV:headAttr:headV`:
  Creates a new edge instance.

- (long long) - `countNodes`:
  Gets the number of nodes.

- (long long) - `countEdges`:
  Gets the number of edges.

- (STSEdgeData *) - `getEdgeData`:
  Gets information about an edge.

- (long long) - `getEdgePeer:node`:
  Gets the other end for the given edge.

- (void) - `drop`:
  Drops the given OID.

- (void) - `dropWithObjects`:
  Drops all the OIDs from the given collection.

- (int) - `getObjectType`:
  Gets the Sparksee node or edge type identifier for the given OID.

- (int) - `createAttribute:name:dt:kind`:
  Creates a new attribute.

- (int) - `createAttributeWithDefault:name:dt:kind:defaultValue`:
  Creates a new attribute with a default value.

- (int) - `createSessionAttribute:dt:kind`:
  Creates a new Session attribute.

- (int) - `createSessionAttributeWithDefault:dt:kind:defaultValue`:
  Creates a new Session attribute with a default value.

- (void) - `setAttributeDefaultValue:value`:
  Sets a default value for an attribute.

- (void) - `setAttributeText:attr:tstream`:
  Sets the writable TextStream for the given text attribute and OID.

- (void) - `setAttribute:attr:value`:
  Sets the Value for the given attribute and OID.

- (STSAttributeStatistics *) - `getAttributeStatistics:basic`:
  Gets the read-only TextStream for the given text attribute and OID.

- (void) - `setAttributeText:attr:tstream`:
  Sets the writable TextStream for the given text attribute and OID.

- (void) - `setAttribute:attr:value`:
  Sets the Value for the given attribute and OID.
Gets statistics from the given attribute.

- (long long) - getAttributeIntervalCount:lower:includeLower:higher:includeHigher:
  Gets how many objects have a value into the given range for the given attribute.

- (int) - findType:
  Gets the Sparksee type identifier for the given type name.

- (STSType *) - getType:
  Gets information about the given type.

- (void) - removeType:
  Removes the given type.

- (void) - renameTypeWithName:newName:
  Renames a type.

- (void) - renameType:newName:
  Renames a type.

- (long long) - findObject:value:
  Finds one object having the given Value for the given attribute.

- (long long) - findOrCreateObject:value:
  Finds one object having the given Value for the attribute or it creates one does not exist any.

- (STSObjects *) - selectWithType:
  Selects all OIDs belonging to the given type.

- (STSObjects *) - selectWithAttrValue:cond:value:
  Selects all OIDs satisfying the given condition for the given attribute.

- (STSObjects *) - selectWithAttrValues:cond:lower:higher:
  Selects all OIDs satisfying the given condition for the given attribute.

- (STSObjects *) - selectWithAttrValueRestriction:cond:value:restriction:
  Selects all OIDs satisfying the given condition for the given attribute.

- (STSObjects *) - selectWithAttrValuesRestriction:cond:lower:higher:restriction:
  Selects all OIDs satisfying the given condition for the given attribute.

- (STSObjects *) - explode:etype:dir:
  Selects all edges from or to the given node OID and for the given edge type.

- (STSObjects *) - explodeWithObjects:etype:dir:
  Selects all edges from or to each of the node OID in the given collection and for the given edge type.

- (long long) - degree:etype:dir:
  Gets the number of edges from or to the given node OID and for the given edge type.

- (STSObjects *) - neighbors:etype:dir:
  Selects all neighbor nodes from or to the given node OID and for the given edge type.

- (STSObjects *) - neighborsWithObjects:etype:dir:
  Selects all neighbor nodes from or to each of the node OID in the given collection and for the given edge type.

- (STSObjects *) - edges:tail:head:
  Gets all the edges of the given type between two given nodes (tail and head).

- (long long) - findEdge:tail:head:
  Gets any of the edges of the given type between two given nodes (tail and head).

- (long long) - findOrCreateEdge:tail:head:
  Gets any of the edges of the specified type between two given nodes (tail and head).
• (STSObjects ∗) - tails:  
  Gets all the tails from the given edges collection.
• (STSObjects ∗) - heads:  
  Gets all the heads from the given edges collection.
• (void) - tailsAndHeads:tails:heads:  
  Gets all the tails and heads from the given edges collection.
• (STSTypeList ∗) - findNodeTypes  
  Gets all existing Sparksee node type identifiers.
• (STSTypeList ∗) - findEdgeTypes  
  Gets all existing Sparksee edge type identifiers.
• (STSTypeList ∗) - findTypes  
  Gets all existing Sparksee node and edge type identifiers.
• (STSAttributeList ∗) - findAttributes:  
  Gets all existing Sparksee attribute identifiers for the given type identifier.
• (STSAttributeList ∗) - getAttributes:  
  Gets all Sparksee attribute identifiers with a non-NULL value for the given Sparksee OID.
• (STSValues ∗) - getValues:  
  Gets the Value collection for the given attribute.
• (void) - dumpData:  
  Dumps logical data to a file.
• (void) - dumpStorage:  
  Dumps internal storage data to a file.
• (void) - exportGraph:type:em:  
  Exports the Graph.
• (void) - backup:  
  Dumps all the data to a backup file.

3.24.1 Detailed Description

Graph class.
Each Database has a Graph associated, which is the persistent graph which contains all data stored into the graph database and is retrieved from a Session.

Check out the 'API' and the 'SPARKSEE graph database' sections in the SPARKSEE User Manual for more details on the use of this class.

Author
Sparsity Technologies http://www.sparsity-technologies.com

3.24.2 Method Documentation

3.24.2.1 - (void) backup: (NSString ∗) file

Dumps all the data to a backup file.
See the Sparksee class Restore method.

Parameters

| file | [in] Output backup file path. |
Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.IO.IOException</td>
<td>If the given file cannot be created.</td>
</tr>
<tr>
<td>System.Application-</td>
<td>null</td>
</tr>
<tr>
<td>Exception</td>
<td></td>
</tr>
</tbody>
</table>

3.24.2.2 - (long long) countEdges

Gets the number of edges.

Returns

The number of edges.

3.24.2.3 - (long long) countNodes

Gets the number of nodes.

Returns

The number of nodes.

3.24.2.4 - (int) createAttribute: (int) type: (int) name: (NSString *) name: (enum STSDataType) dt: (enum STSAttributeKind) kind

Creates a new attribute.

Parameters

- `type` [in] Sparksee node or edge type identifier.
- `name` [in] Unique name for the new attribute.
- `dt` [in] Data type for the new attribute.
- `kind` [in] Attribute kind.

Returns

Unique Sparksee attribute identifier.

3.24.2.5 - (int) createAttributeWithDefault: (int) type: (int) name: (NSString *) name: (enum STSDataType) dt: (enum STSAttributeKind) kind defaultValue: (STSValue *) defaultValue

Creates a new attribute with a default value.

Parameters

- `type` [in] Sparksee node or edge type identifier.
- `name` [in] Unique name for the new attribute.
- `dt` [in] Data type for the new attribute.
- `kind` [in] Attribute kind.
- `defaultValue` [in] The default value to use in each new node/edge.

Returns

Unique Sparksee attribute identifier.

3.24.2.6 - (long long) createEdge: (int) type: (long long) tail: (long long) head

Creates a new edge instance.
Parameters

<table>
<thead>
<tr>
<th>type</th>
<th>[in] Sparksee type identifier.</th>
</tr>
</thead>
<tbody>
<tr>
<td>tail</td>
<td>[in] Source Sparksee OID.</td>
</tr>
<tr>
<td>head</td>
<td>[in] Target Sparksee OID.</td>
</tr>
</tbody>
</table>

Returns

Unique OID of the new edge instance.

3.24.2.7 - (int) createEdgeType: (NSString *) name directed:(BOOL) directed neighbors:(BOOL) neighbors

Creates a new edge type.

Parameters

<table>
<thead>
<tr>
<th>name</th>
<th>[in] Unique name for the new edge type.</th>
</tr>
</thead>
<tbody>
<tr>
<td>directed</td>
<td>[in] If TRUE, this creates a directed edge type, otherwise this creates a undirected edge type.</td>
</tr>
<tr>
<td>neighbors</td>
<td>[in] If TRUE, this indexes neighbor nodes, otherwise not.</td>
</tr>
</tbody>
</table>

Returns

Unique Sparksee type identifier.

3.24.2.8 - (long long) createEdgeWithAttributes: (int) type tailAttr:(int) tailAttr tailV:(STSValue *) tailV headAttr:(int) headAttr headV:(STSValue *) headV

Creates a new edge instance.

The tail of the edge will be any node having the given tailV Value for the given tailAttr attribute identifier, and the head of the edge will be any node having the given headV Value for the given headAttr attribute identifier.

Parameters

<table>
<thead>
<tr>
<th>type</th>
<th>[in] Sparksee type identifier.</th>
</tr>
</thead>
<tbody>
<tr>
<td>headAttr</td>
<td>[in] Sparksee attribute identifier.</td>
</tr>
<tr>
<td>headV</td>
<td>[in] Value.</td>
</tr>
</tbody>
</table>

Returns

Unique OID of the new edge instance.

3.24.2.9 - (int) createNodeType: (NSString *) name

Creates a new node type.

Parameters

| type  | [in] Sparksee type identifier. |

Returns

Unique OID of the new node instance.

3.24.2.10 - (int) createNodeType: (NSString *) name

Creates a new node type.
### Parameters

| name       | [in] Unique name for the new node type. |

### Returns

Unique Sparksee type identifier.

#### 3.24.2.11 - (int) createRestrictedEdgeType: (NSString ∗) name tail:(int) tail head:(int) head neighbors:(BOOL) neighbors

Creates a new restricted edge type.

| name       | [in] Unique name for the new edge type. |
| tail       | [in] Tail Sparksee node type identifier. |
| head       | [in] Head Sparksee node type identifier. |
| neighbors  | [in] If TRUE, this indexes neighbor nodes, otherwise not. |

### Returns

Unique Sparksee type identifier.

#### 3.24.2.12 - (int) createSessionAttribute: (int) type dt:(enum STSDataType) dt kind:(enum STSAttributeKind) kind

Creates a new Session attribute.

Session attributes are exclusive for the Session (just its Session can use the attribute) and are automatically removed when the Session is closed (thus, attribute data is not persistent into the database).

Since they are not persistent, they cannot be retrieved from the database, so they do not have an identifier name.

| type       | [in] Sparksee node or edge type identifier. |
| dt         | [in] Data type for the new attribute. |
| kind       | [in] Attribute kind. |

### Returns

Unique Sparksee attribute identifier.

#### 3.24.2.13 - (int) createSessionAttributeWithDefault: (int) type dt:(enum STSDataType) dt kind:(enum STSAttributeKind) kind defaultvalue:(STSValue ∗) defaultvalue

Creates a new Session attribute with a default value.

Session attributes are exclusive for the Session (just its Session can use the attribute) and are automatically removed when the Session is closed (thus, attribute data is not persistent into the database).

Since they are not persistent, they cannot be retrieved from the database, so they do not have an identifier name.

| type       | [in] Sparksee node or edge type identifier. |
| dt         | [in] Data type for the new attribute. |
| kind       | [in] Attribute kind. |
| defaultvalue | [in] The default value to use in each new node/edge. |
Returns

Unique Sparksee attribute identifier.

3.24.2.14  - (long long) degree: (long long) oid etype:(int) etype dir:(enum STSEdgesDirection) dir

Gets the number of edges from or to the given node OID and for the given edge type.

Parameters

| oid | [in] Sparksee node OID. |
| etype | [in] Sparksee edge type identifier. |
| dir | [in] Direction. |

Returns

The number of edges.

3.24.2.15  - (void) drop: (long long) oid

Drops the given OID.

It also removes its egdges as well as its attribute values.

Parameters

| oid | [in] Sparksee OID to be removed. |

3.24.2.16  - (void) dropWithObjects: (STSObjects ∗) objs

Drops all the OIDs from the given collection.

See Drop method with the single OID parameter. This performs that call for all the elements into the collection.

Parameters

| objs | [in] Objects collection with the OIDs to be removed. |

3.24.2.17  - (void) dumpData: (NSString ∗) file

Dumps logical data to a file.

Parameters

| file | [in] Output file path. |

Exceptions

| System.IO.IOException | If the given file cannot be created. |
| System.ApplicationException | null |

3.24.2.18  - (void) dumpStorage: (NSString ∗) file

Dumps internal storage data to a file.
3.24 STSGraph Class Reference

Parameters

<table>
<thead>
<tr>
<th>name</th>
<th>type</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>file</td>
<td>[in]</td>
<td>Output file path.</td>
</tr>
</tbody>
</table>

Exceptions

<table>
<thead>
<tr>
<th>exception</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.IO.IOException</td>
<td>If the given file cannot be created.</td>
</tr>
<tr>
<td>System.ApplicationException</td>
<td>null</td>
</tr>
</tbody>
</table>

3.24.2.19 - (STSObejcts ∗) edges: (int) etype tail:(long long) tail head:(long long) head

Gets all the edges of the given type between two given nodes (tail and head).

Parameters

<table>
<thead>
<tr>
<th>name</th>
<th>type</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>etype</td>
<td>[in]</td>
<td>Sparksee edge type identifier.</td>
</tr>
<tr>
<td>tail</td>
<td>[in]</td>
<td>Tail node identifier.</td>
</tr>
<tr>
<td>head</td>
<td>[in]</td>
<td>Head node identifier.</td>
</tr>
</tbody>
</table>

Returns

Objects instance.

3.24.2.20 - (STSObejcts ∗) explode: (long long) oid etype:(int) etype dir:(enum STSEdgesDirection) dir

Selects all edges from or to the given node OID and for the given edge type.

Parameters

<table>
<thead>
<tr>
<th>name</th>
<th>type</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>oid</td>
<td>[in]</td>
<td>Sparksee node OID.</td>
</tr>
<tr>
<td>etype</td>
<td>[in]</td>
<td>Sparksee edge type identifier.</td>
</tr>
<tr>
<td>dir</td>
<td>[in]</td>
<td>Direction.</td>
</tr>
</tbody>
</table>

Returns

Objects instance.

3.24.2.21 - (STSObejcts ∗) explodeWithObjects: (STSObejcts ∗) objs etype:(int) etype dir:(enum STSEdgesDirection) dir

Selects all edges from or to each of the node OID in the given collection and for the given edge type.

Parameters

<table>
<thead>
<tr>
<th>name</th>
<th>type</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>objs</td>
<td>[in]</td>
<td>Sparksee node OID collection.</td>
</tr>
<tr>
<td>etype</td>
<td>[in]</td>
<td>Sparksee edge type identifier.</td>
</tr>
<tr>
<td>dir</td>
<td>[in]</td>
<td>Direction.</td>
</tr>
</tbody>
</table>

Returns

Objects instance.

3.24.2.22 - (void) exportGraph: (NSString ∗) file type:(enum STSExportType) type em:(STSExportManager ∗) em

Exports the Graph.
3.24 STSGraph Class Reference

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>[in] Export type.</td>
</tr>
<tr>
<td>em</td>
<td>[in] Defines how to do the export for each graph object.</td>
</tr>
</tbody>
</table>

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.IO.IOException</td>
<td>null</td>
</tr>
</tbody>
</table>

3.24.2.23 - (int) findAttribute: (int) type name:(NSString *) name

Gets the Sparksee attribute identifier for the given type identifier and attribute name.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>[in] Sparksee type identifier.</td>
</tr>
<tr>
<td>name</td>
<td>[in] Unique attribute name.</td>
</tr>
</tbody>
</table>

Returns

The Sparksee attribute identifier for the given type and attribute name or InvalidAttribute if there is no attribute with the given name for the given type.

3.24.2.24 - (STSAttributeList *) findAttributes: (int) type

Gets all existing Sparksee attribute identifiers for the given type identifier.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>[in] Sparksee type identifier.</td>
</tr>
</tbody>
</table>

Returns

Sparksee attribute identifier list.

3.24.2.25 - (long long) findEdge: (int) etype tail:(long long) tail head:(long long) head

Gets any of the edges of the given type between two given nodes (tail and head).

If there are more than one, then any of them will be returned. And in case there are no edge between the given tail and head, the Objects InvalidOID will be returned.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>etype</td>
<td>[in] Sparksee edge type identifier.</td>
</tr>
<tr>
<td>tail</td>
<td>[in] Tail node identifier.</td>
</tr>
<tr>
<td>head</td>
<td>[in] Head node identifier.</td>
</tr>
</tbody>
</table>

Returns

Any of the edges or the Objects InvalidOID.

3.24.2.26 - (STSTypeList *) findEdgeTypes

Gets all existing Sparksee edge type identifiers.
Returns
Sparksee edge type identifier list.

3.24.2.27 - (STSTypeList *) findNodeTypes

Gets all existing Sparksee node type identifiers.

Returns
Sparksee node type identifier list.

3.24.2.28 - (long long) findObject: (int) attr value:(STSValue *) value

Finds one object having the given Value for the given attribute.
If there are more than one, then any of them will be returned. And in case there are no object having this Value, the
Objects InvalidOID will be returned.

Parameters

<table>
<thead>
<tr>
<th>attr</th>
<th>[in] Sparksee attribute identifier.</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>[in] Value.</td>
</tr>
</tbody>
</table>

Returns
Sparksee OID or the Objects InvalidOID.

3.24.2.29 - (long long) findOrCreateEdge: (int) etype tail:(long long) tail head:(long long) head

Gets any of the edges of the specified type between two given nodes (tail and head).
If it can not find any edge of this type between them it tries to create a new one.

Parameters

<table>
<thead>
<tr>
<th>etype</th>
<th>[in] Sparksee edge type identifier.</th>
</tr>
</thead>
<tbody>
<tr>
<td>tail</td>
<td>[in] Tail node identifier.</td>
</tr>
<tr>
<td>head</td>
<td>[in] Head node identifier.</td>
</tr>
</tbody>
</table>

Returns
Any of the edges or the Objects InvalidOID.

3.24.2.30 - (long long) findOrCreateObject: (int) attr value:(STSValue *) value

Finds one object having the given Value for the attribute or it creates one does not exist any.
If the attribute is a node or edge attribute and at least one node/edge with that value is found, it returns one of them.
But if it does not exist, then: If it’s a node attribute it will create it and set the attribute. If it’s an edge attribute it will
return the InvalidOID.
Using this method with a global attribute will return the InvalidOID.

Parameters

<table>
<thead>
<tr>
<th>attr</th>
<th>[in] Sparksee attribute identifier.</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>[in] Value.</td>
</tr>
</tbody>
</table>
Returns
Sparksee OID or the Objects InvalidOID.

3.24.2.31 - (int) findType: (NSString *) name

Gets the Sparksee type identifier for the given type name.

Parameters

| name | [in] Unique type name. |

Returns
The Sparksee type identifier for the given type name or the Type InvalidType if there is no type with the given name.

3.24.2.32 - (STSTypeList *) findTypes

Gets all existing Sparksee node and edge type identifiers.

Returns
Sparksee node and edge type identifier list.

3.24.2.33 - (STSAttribute *) getAttribute: (int) attr

Gets information about the given attribute.

Parameters

| attr | [in] Sparksee attribute identifier. |

Returns
The Attribute for the given Sparksee attribute identifier.

3.24.2.34 - (long long) getAttributeIntervalCount: (int) attr lower:(STSValue *) lower includeLower:(BOOL) includeLower higher:(STSValue *) higher includeHigher:(BOOL) includeHigher

Gets how many objects have a value into the given range for the given attribute.
This only works for the attributes with the AttributeKind Indexed or Unique.
Given values must belong to the same DataType than the attribute.

Parameters

| attr | [in] Sparksee attribute identifier. |
| lower | [in] Lower bound Value of the range. |
| includeLower | [in] If TRUE, include lower bound Value of the range. |
| higher | [in] Higher bound Value of the range. |
| includeHigher | [in] If TRUE, include higher bound Value of the range. |

Returns
Number of objects having a value into the given range.
3.24.2.35 - (void) getAttributeInValue: (long long) oid attr:(int) attr value: (STSValue ∗) value

Gets the Value for the given attribute and OID.

Parameters

<table>
<thead>
<tr>
<th>oid</th>
<th>[in] Sparksee OID.</th>
</tr>
</thead>
<tbody>
<tr>
<td>attr</td>
<td>[in] Sparksee attribute identifier.</td>
</tr>
<tr>
<td>value</td>
<td>[in</td>
</tr>
</tbody>
</table>

3.24.2.36 - (STSAttributeList ∗) getAttributes: (long long) oid

Gets all Sparksee attribute identifiers with a non-NULL value for the given Sparksee OID.

Parameters

| oid          | [in] Sparksee OID. |

Returns

Sparksee attribute identifier list.

3.24.2.37 - (STSAttributeStatistics ∗) getAttributeStatistics: (int) attr basic:(BOOL) basic

Gets statistics from the given attribute.

Parameters

| attr       | [in] Sparksee attribute identifier. |
| basic      | [in] If FALSE all statistics are computed, if TRUE just those statistics marked as basic will be computed (see description of the AttributeStatistics class). Of course, computing just basic statistics will be faster than computing all of them. |

Returns

An AttributeStatistics instance.

3.24.2.38 - (STSTextStream ∗) getAttributeText: (long long) oid attr:(int) attr

Gets the read-only TextStream for the given text attribute and OID.

Parameters

| oid          | [in] Sparksee OID. |
| attr         | [in] Sparksee attribute identifier. |

Returns

A TextStream. This returns a TextStream to read.

3.24.2.39 - (STSValue ∗) getAttributeWithOid: (long long) oid attr:(int) attr

Gets the Value for the given attribute and OID.

The other version of this call, where the Value is an output parameter instead of the return, is better because it allows the user to reuse an existing Value instance, whereas this call always creates a new Value instance to be returned.

It never returns NULL. Thus, in case the OID has a NULL value for the attribute it returns a NULL Value instance.
Parameters

| oid | [in] Sparksee OID. |
| attr | [in] Sparksee attribute identifier. |

Returns

A new Value instance having the attribute value for the given OID.

3.24.2.40 - (STSEdgeData*) getEdgeData: (long long) edge

Gets information about an edge.

Parameters

| edge | [in] Sparksee edge identifier. |

Returns

An EdgeData instance.

3.24.2.41 - (long long) getEdgePeer: (long long) edge node:(long long) node

Gets the other end for the given edge.

Parameters

| edge | [in] Sparksee edge identifier. |
| node | [in] Sparksee node identifier. It must be one of the ends of the edge. |

Returns

The other end of the edge.

3.24.2.42 - (int) getObjectType: (long long) oid

Gets the Sparksee node or edge type identifier for the given OID.

Parameters

| oid | [in] Sparksee OID. |

Returns

Sparksee node or edge type identifier.

3.24.2.43 - (STSType*) getType: (int) type

Gets information about the given type.

Parameters

| type | [in] Sparksee type identifier. |

Returns

The Type for the given Sparksee type identifier.
3.24.2.44 - (STSValues *) getValues: (int) attr

Gets the Value collection for the given attribute.

**Parameters**

| attr | [in] Sparksee attribute identifier. |

**Returns**

Returns a Values object.

3.24.2.45 - (STSObjects *) heads: (STSObjects *) edges

Gets all the heads from the given edges collection.

**Parameters**

| edges | [in] Sparksee edge identifier collection. |

**Returns**

The heads collection.

3.24.2.46 - (void) indexAttribute: (int) attr kind:(enum STSAttributeKind) kind

Updates the index of the given attribute.

This just works if the current index of the attribute corresponds to the AttributeKind Basic and the new one is Indexed or Unique.

**Parameters**

| attr | [in] Sparksee attribute identifier. |
| kind | [in] Attribute kind. |

3.24.2.47 - (STSObjects *) neighbors: (long long) oid etype:(int) etype dir:(enum STSEdgesDirection) dir

Selects all neighbor nodes from or to the given node OID and for the given edge type.

**Parameters**

| oid  | [in] Sparksee node OID. |
| etype| [in] Sparksee edge type identifier. |
| dir  | [in] Direction. |

**Returns**

Objects instance.

3.24.2.48 - (STSObjects *) neighborsWithObjects: (STSObjects *) objs etype:(int) etype dir:(enum STSEdgesDirection) dir

Selects all neighbor nodes from or to each of the node OID in the given collection and for the given edge type.

**Parameters**

| objs | [in] Sparksee node OID collection. |
| etype| [in] Sparksee edge type identifier. |
| dir  | [in] Direction. |
Returns
   Objects instance.

3.24.2.49  - (void) removeAttribute: (int) attr

Removes the given attribute.

Parameters

| attr | [in] Sparksee attribute identifier. |

3.24.2.50  - (void) removeType: (int) type

Removes the given type.
This fails if there exist attributes defined for the type or if there exist restricted edges referencing this type.

Parameters

| type | [in] Sparksee type identifier. |

3.24.2.51  - (void) renameAttribute: (int) attr newName:(NSString *) newName

Renames an attribute.
The new name must be available.

Parameters

| attr | [in] Sparksee attribute identifier. |
| newName | [in] The new name for the attribute. |

3.24.2.52  - (void) renameType: (int) type newName:(NSString *) newName

Renames a type.
The new name must be available.

Parameters

| type | [in] The type to be renamed. |
| newName | [in] The new name for the type. |

3.24.2.53  - (void) renameTypeWithName: (NSString *) oldName newName:(NSString *) newName

Renames a type.
The new name must be available.

Parameters

| oldName | [in] The current name of the type to be renamed. |
| newName | [in] The new name for the type. |

3.24.2.54  - (STSObjects *) selectWithAttrValue: (int) attr cond:(enum STSCondition) cond value:(STSValue *) value

Selects all OIDs satisfying the given condition for the given attribute.
Parameters

<table>
<thead>
<tr>
<th>attr</th>
<th>[in] Sparksee attribute identifier.</th>
</tr>
</thead>
<tbody>
<tr>
<td>cond</td>
<td>[in] Condition to be satisfied.</td>
</tr>
<tr>
<td>value</td>
<td>[in] Value to be satisfied.</td>
</tr>
</tbody>
</table>

Returns

Objects instance.

3.24.2.55 - (STSObjects *) selectWithAttrValueRestriction: (int) attr cond:(enum STSCondition) cond value:(STSValue *)
value restriction:(STSObjects *) restriction

Selects all OIDs satisfying the given condition for the given attribute.

Parameters

<table>
<thead>
<tr>
<th>attr</th>
<th>[in] Sparksee attribute identifier.</th>
</tr>
</thead>
<tbody>
<tr>
<td>cond</td>
<td>[in] Condition to be satisfied.</td>
</tr>
<tr>
<td>value</td>
<td>[in] Value to be satisfied.</td>
</tr>
<tr>
<td>restrict</td>
<td>[in] Objects to limit the select in this set of objects.</td>
</tr>
</tbody>
</table>

Returns

Objects instance.

3.24.2.56 - (STSObjects *) selectWithAttrValues: (int) attr cond:(enum STSCondition) cond lower:(STSValue *) lower
higher:(STSValue *) higher

Selects all OIDs satisfying the given condition for the given attribute. This allows to perform the Between operation, thus it has two Value arguments.

Parameters

<table>
<thead>
<tr>
<th>attr</th>
<th>[in] Sparksee attribute identifier.</th>
</tr>
</thead>
<tbody>
<tr>
<td>cond</td>
<td>[in] Condition to be satisfied.</td>
</tr>
<tr>
<td>lower</td>
<td>[in] Lower-bound Value to be satisfied.</td>
</tr>
<tr>
<td>higher</td>
<td>[in] Higher-bound Value to be satisfied.</td>
</tr>
</tbody>
</table>

Returns

Objects instance.

3.24.2.57 - (STSObjects *) selectWithAttrValuesRestriction: (int) attr cond:(enum STSCondition) cond lower:(STSValue *)
lower higher:(STSValue *) higher restriction:(STSObjects *) restriction

Selects all OIDs satisfying the given condition for the given attribute. This allows to perform the Between operation, thus it has two Value arguments.

Parameters

<table>
<thead>
<tr>
<th>attr</th>
<th>[in] Sparksee attribute identifier.</th>
</tr>
</thead>
<tbody>
<tr>
<td>cond</td>
<td>[in] Condition to be satisfied.</td>
</tr>
<tr>
<td>lower</td>
<td>[in] Lower-bound Value to be satisfied.</td>
</tr>
<tr>
<td>higher</td>
<td>[in] Higher-bound Value to be satisfied.</td>
</tr>
<tr>
<td>restrict</td>
<td>[in] Objects to limit the select in this set of objects.</td>
</tr>
</tbody>
</table>
3.24 STSGraph Class Reference

Returns

Objects instance.

3.24.2.58 - (STSObjects *) selectWithType: (int) type

Selects all OIDs belonging to the given type.

Parameters

| type | [in] Sparksee type identifier. |

Returns

Objects instance.

3.24.2.59 - (void) setAttribute: (long long) oid attr:(int) attr value: (STSValue *) value

Sets the Value for the given attribute and OID.

Parameters

| oid  | [in] Sparksee OID. |
| attr | [in] Sparksee attribute identifier. |
| value | [in] Value for the given attribute and for the given OID. |

3.24.2.60 - (void) setAttributeDefaultValue: (int) attr value: (STSValue *) value

Sets a default value for an attribute.

The default value will be applied to all the new nodes or edges.

The given value must have the same DataType as the attribute or be a NULL value to remove the current default value.

Parameters

| value | [in] The default value to use for this attribute. |

3.24.2.61 - (void) setAttributeText: (long long) oid attr:(int) attr tstream: (STSTextStream *) tstream

Sets the writable TextStream for the given text attribute and OID.

Parameters

| oid  | [in] Sparksee OID. |
| attr | [in] Sparksee attribute identifier. |
| tstream | [in] New Text value. This corresponds to a TextStream to write. |

3.24.2.62 - (STSObjects *) tails: (STSObjects *) edges

Gets all the tails from the given edges collection.

Parameters

| edges | [in] Sparksee edge identifier collection. |
3.24.2.63 - (void) tailsAndHeads: (STSObjects *) edges tails:(STSObjects *) tails heads:(STSObjects *) heads

Gets all the tails and heads from the given edges collection.

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>edges</td>
<td>[in] Sparksee edge identifier collection.</td>
</tr>
<tr>
<td>tails</td>
<td>[in</td>
</tr>
<tr>
<td>heads</td>
<td>[in</td>
</tr>
</tbody>
</table>

The documentation for this class was generated from the following file:

- Sparksee.h

3.25 STSGraphExport Class Reference

Stores the graph exporting values.

Inheritance diagram for STSGraphExport:

```
     NSObject

     +-----------------+
     |                 |
     | STSGraphExport  |
     +-----------------+
```

Collaboration diagram for STSGraphExport:

```
     NSObject

     +-----------------+
     |                 |
     | STSGraphExport  |
     +-----------------+
```
Instance Methods

- (id) - init
  Creates a new GraphExport instance.

- (void) - setDefaults
  Sets to default values.

- (NSString *) - getLabel
  Gets the graph label.

- (void) - setLabel:
  Sets the graph label.

3.25.1 Detailed Description

Stores the graph exporting values.

Author

Sparsity Technologies [http://www.sparsity-technologies.com](http://www.sparsity-technologies.com)

3.25.2 Method Documentation

3.25.2.1 - (NSString *) getLabel

Gets the graph label.

Returns

The graph label.

3.25.2.2 - (void) setLabel: (NSString *) label

Sets the graph label.

Parameters

| label | [in] The graph label. |

The documentation for this class was generated from the following file:

- Sparksee.h

3.26 STSInt32List Class Reference

Sparksee 32-bit signed integer list.
Inheritance diagram for STSInt32List:

Collaboration diagram for STSInt32List:

Instance Methods

- (int) - count
  Number of elements in the list.
- (id) - init
  Constructor.
- (void) - add:
  Adds an 32-bit signed integer at the end of the list.
- (void) - clear
  Clears the list.
- (id) - initWithArray:
  Creates a new Int32List instance from the given array.
- (id) - initWithNSEnumerator:
  Creates a new Int32List instance from the given NSEnumerator.
- (STSInt32ListIterator *) - iterator
  Gets a new Int32ListIterator.
3.26.1 Detailed Description

Sparksee 32-bit signed integer list.
It stores a 32-bit signed integer list.
Use Int32ListIterator to access all elements into this collection.

Author

Sparsity Technologies http://www.sparsity-technologies.com

3.26.2 Method Documentation

3.26.2.1 - (void) add: (int) value

Adds an 32-bit signed integer at the end of the list.

Parameters

| value | [in] The integer. |

3.26.2.2 - (int) count

Number of elements in the list.

Returns

Number of elements in the list.

3.26.2.3 - (id) init

Constructor.
This creates an empty list.
The documentation for this class was generated from the following file:

• Sparksee.h

3.27 STSInt32ListIterator Class Reference

Int32List iterator class.

Inheritance diagram for STSInt32ListIterator:

[Diagram of inheritance]
Collaboration diagram for STSInt32ListIterator:

```
+----------------+      +----------------+      +----------------+
| STSInt32List   |      | NSObject       |      | STSInt32ListIterator |
+----------------+      +----------------+      +----------------+
     |                        |      |                  |
     |                        |      |                  |
     +----------------+      +----------------+      +----------------+
                 |                        |      |                  |
                 |                        |      |                  |
                 +----------------+      +----------------+      +----------------+
                  |                        |      |                  |
                  |                        |      |                  |
                  +----------------+      +----------------+      +----------------+
                        |                  |
                        |                  |
                        +----------------+
                           | theParent |
                           +-----------+

Instance Methods

• (int) - next
  Moves to the next element.

• (BOOL) - hasNext
  Gets if there are more elements.

3.27.1 Detailed Description

Int32List iterator class.
Iterator to traverse all the integer into a Int32List instance.

Author

Sparsity Technologies http://www.sparsity-technologies.com

3.27.2 Method Documentation

3.27.2.1 - (BOOL) hasNext

Gets if there are more elements.

Returns

  TRUE if there are more elements, FALSE otherwise.

3.27.2.2 - (int) next

Moves to the next element.
Returns

The next element.

The documentation for this class was generated from the following file:

- Sparksee.h

### 3.28 STSNodeExport Class Reference

Stores the node exporting values.

Inheritance diagram for STSNodeExport:

![Inheritance Diagram](image)

Collaboration diagram for STSNodeExport:

![Collaboration Diagram](image)

### Instance Methods

- (id) - `init`
  *Creates a new instance.*
- (void) - `setDefaults`
  *Sets to default values.*
- (NSString *) - `setLabel`
  *Gets the node label.*
- (void) - `setLabel:`
Sets the node label.
• (enum STSNodeShape) - getShape
  Gets the node shape.
• (void) - setShape:
  Sets the node shape.
• (int) - getColorRGB
  Gets the node color.
• (void) - setColorRGB:
  Sets the node color.
• (int) - getLabelColorRGB
  Gets the node label color.
• (void) - setLabelColorRGB:
  Sets the node label color.
• (int) - getHeight
  Gets the node height.
• (void) - setHeight:
  Sets the node height.
• (int) - getWidth
  Gets the node width.
• (void) - setWidth:
  Gets the node width.
• (BOOL) - isFit
  Gets whether the node size is fitted to the label or not.
• (void) - setFit:
  Sets the node fit property.
• (int) - getFontSize
  Gets the node label font size.
• (void) - setFontSize:
  Sets the node label font size.
• (void) - getColorRed:green:blue:alpha:
  Get the node color separated in RGBA.
• (void) - setColorRed:green:blue:alpha:
  Set the node color with separated RGBA components.
• (void) - getLabelColorRed:green:blue:alpha:
  Get the node label color separated in RGBA.
• (void) - setLabelColorRed:green:blue:alpha:
  Set the node label color with separated RGBA components.

3.28.1 Detailed Description

Stores the node exporting values.
When ‘fit’ is set to TRUE, then ‘height’ and ‘width’ will be ignored.
Some properties may be ignored depending on the exportation type.
Default values are:
Label: "" (empty string).
Shape: Box.
Color: 10863606 (0xa5c3f6).
Label color: 0 (0x000000, Black).
3.28 STSNodeExport Class Reference

Height: 25px.
Width: 25px.
Fit: TRUE.
Font size: 10.

Author
Sparsity Technologies http://www.sparsity-technologies.com

3.28.2 Method Documentation

3.28.2.1 - (void) getColorRed: (double ∗) red green:(double ∗) green blue:(double ∗) blue alpha:(double ∗) alpha

Get the node color separated in RGBA.

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>red</td>
<td>[out]</td>
<td>The red color component ([0..1]).</td>
</tr>
<tr>
<td>green</td>
<td>[out]</td>
<td>The green color component ([0..1]).</td>
</tr>
<tr>
<td>blue</td>
<td>[out]</td>
<td>The blue color component ([0..1]).</td>
</tr>
<tr>
<td>alpha</td>
<td>[out]</td>
<td>The alpha component ([0..1]).</td>
</tr>
</tbody>
</table>

3.28.2.2 - (int) getColorRGB

Gets the node color.

Returns
The node color.

3.28.2.3 - (int) getFontSize

Gets the node label font size.

Returns
The node label font size.

3.28.2.4 - (int) getHeight

Gets the node height.

Returns
The node height in pixels.

3.28.2.5 - (NSString ∗) getLabel

Gets the node label.

Returns
The node label.

3.28.2.6 - (void) getLabelColorRed: (double ∗) red green:(double ∗) green blue:(double ∗) blue alpha:(double ∗) alpha

Get the node label color separated in RGBA.
3.28 STSNodeExport Class Reference

Parameters

<table>
<thead>
<tr>
<th>red</th>
<th>[out] The red color component ([0..1]).</th>
</tr>
</thead>
<tbody>
<tr>
<td>green</td>
<td>[out] The green color component ([0..1]).</td>
</tr>
<tr>
<td>blue</td>
<td>[out] The blue color component ([0..1]).</td>
</tr>
<tr>
<td>alpha</td>
<td>[out] The alpha component ([0..1]).</td>
</tr>
</tbody>
</table>

3.28.2.7 - (int) getLabelColorRGB

Gets the node label color.

Returns

The node label color.

3.28.2.8 - (enum STSNodeShape) getShape

Gets the node shape.

Returns

The node shape.

3.28.2.9 - (int) getWidth

Gets the node width.

Returns

The node width in pixels.

3.28.2.10 - (BOOL) isFit

Gets whether the node size is fitted to the label or not.

Returns

If TRUE, then the node size is fitted to the label, otherwise the size is fixed with the values of 'height' and 'width'.

3.28.2.11 - (void) setColorRed: (double) red green:(double) green blue:(double) blue alpha:(double) alpha

Set the node color with separated RGBA components.

Parameters

<table>
<thead>
<tr>
<th>red</th>
<th>[in] The red color component ([0..1]).</th>
</tr>
</thead>
<tbody>
<tr>
<td>green</td>
<td>[in] The green color component ([0..1]).</td>
</tr>
<tr>
<td>blue</td>
<td>[in] The blue color component ([0..1]).</td>
</tr>
<tr>
<td>alpha</td>
<td>[in] The alpha component ([0..1]).</td>
</tr>
</tbody>
</table>

3.28.2.12 - (void) setColorRGB: (int) color

Sets the node color.

Parameters

<table>
<thead>
<tr>
<th>color</th>
<th>The node color.</th>
</tr>
</thead>
</table>

Generated on Tue Jun 2 2015 15:49:36 for Sparksee by Doxygen
3.28.2.13 - (void) setFit: (BOOL) fit

Sets the node fit property.

Parameters

fit [in] If TRUE, then the node size is fitted to the label ('height' and 'width' will be ignored), otherwise the size is fixed with the values of 'height' and 'width'.

3.28.2.14 - (void) setFontSize: (int) size

Sets the node label font size.

Parameters

size [in] The node label font size.

3.28.2.15 - (void) setHeight: (int) height

Sets the node height.

Parameters

height [in] The node height in pixels.

3.28.2.16 - (void) setLabel: (NSString *) label

Sets the node label.

Parameters

label [in] The node label.

3.28.2.17 - (void) setLabelColorRed: (double) red green:(double) green blue:(double) blue alpha:(double) alpha

Set the node label color with separated RGBA components.

Parameters

red [in] The red color component ([0..1]).
green [in] The green color component ([0..1]).
blue [in] The blue color component ([0..1]).
alpha [in] The alpha component ([0..1]).

3.28.2.18 - (void) setLabelColorRGB: (int) color

Sets the node label color.

Parameters

color [in] The node label color.

3.28.2.19 - (void) setShape: (enum STSNodeShape) shape

Sets the node shape.
3.29 STSNodeTypeExporter Class Reference

Parameters

| shape | [in] The node shape. |

3.28.2.20 - (void) setWidth: (int) width

Gets the node width.

Parameters

| width | The node width in pixels. |

The documentation for this class was generated from the following file:

- Sparksee.h

3.29 STSNodeTypeExporter Class Reference

NodeTypeExporter class.

Inheritance diagram for STSNodeTypeExporter:
Collaboration diagram for STSNodeTypeExporter:

```
+-------------------+                +-------------------+
| STSShortLabelExp  |                | NSObject          |
|                  +-------------------+                +-------------------+
| STSTypeExporter  |                | STSNodeTypeExporter|
|                  +-------------------+                +-------------------+
```

Instance Methods

- `(id)` - `init`
  Creates a new instance.

- `(id)` - `initWithRowWriter:graph:type:attrs:`
  Creates a new instance.

- `(void)` - `run`
  See the TypeExporter class Run method.

- `(void)` - `registerListener`:
  Registers a new listener.

- `(void)` - `setRowWriter`:
  Sets the output data destination.

- `(void)` - `setGraph`:
  Sets the graph that will be exported.

- `(void)` - `setType`:
  Sets the type to be exported.

- `(void)` - `setAttributes`:
  Sets the list of Attributes.

- `(void)` - `setFrequency`:
  Sets the frequency of listener notification.

- `(void)` - `setHeader`:
  Sets the presence of a header row.

### 3.29.1 Detailed Description

NodeTypeExporter class.

Specific TypeExporter implementation for node types.

Check out the 'Data export' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies [http://www.sparsity-technologies.com](http://www.sparsity-technologies.com)
3.29.2 Method Documentation

3.29.2.1 - (id) initWithRowWriter: (STSRowWriter *) rowWriter graph:(STSGraph *) graph type:(int) type attrs:(STSAttributeList *) attrs

Creates a new instance.

Parameters

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>graph</td>
<td>[in] Graph.</td>
</tr>
<tr>
<td>type</td>
<td>[in] Type identifier.</td>
</tr>
<tr>
<td>attrs</td>
<td>[in] Attribute identifiers to be exported.</td>
</tr>
</tbody>
</table>

3.29.2.2 - (void) registerListener: (STSTypeExporterListener *) tel

Registers a new listener.

Parameters

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>tel</td>
<td>[in] TypeExporterListener to be registered.</td>
</tr>
</tbody>
</table>

3.29.2.3 - (void) run

See the TypeExporter class Run method.

Exceptions

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>System.IO.IOException</td>
<td>null</td>
</tr>
<tr>
<td>System.ApplicationException</td>
<td>null</td>
</tr>
</tbody>
</table>

Implements STSTypeExporter.

3.29.2.4 - (void) setAttributes: (STSAttributeList *) attrs

Sets the list of Attributes.

Parameters

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>attrs</td>
<td>[in] Attribute identifiers to be exported</td>
</tr>
</tbody>
</table>

3.29.2.5 - (void) setFrequency: (int) freq

Sets the frequency of listener notification.

Parameters

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>freq</td>
<td>[in] Frequency in number of rows managed to notify progress to all listeners</td>
</tr>
</tbody>
</table>

3.29.2.6 - (void) setGraph: (STSGraph *) graph

Sets the graph that will be exported.

Parameters

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>graph</td>
<td>[in] Graph.</td>
</tr>
</tbody>
</table>
3.29.2.7 - (void) setHeader: (BOOL) header

Sets the presence of a header row.

Parameters

| header          | [in] If TRUE, a header row is dumped with the name of the attributes. |

3.29.2.8 - (void) setRowWriter: (STSTRowWriter *) rw

Sets the output data destination.

Parameters

| rw              | [in] Input RowWriter. |

3.29.2.9 - (void) setType: (int) type

Sets the type to be exported.

Parameters

| type            | [in] Type identifier. |

The documentation for this class was generated from the following file:

- Sparksee.h

3.30 STSNodeTypeLoader Class Reference

NodeTypeLoader class.

Inheritance diagram for STSNodeTypeLoader:
Collaboration diagram for STSNodeTypeLoader:

```
NSObject
    STSTypeLoader
    STSNodeTypeLoader
```

Instance Methods

- (id) - `init`
  Creates a new instance.
- (id) - `initWithRowReader:graph:type:attrs:attrsPos:``
  Creates a new instance.
- (void) - `run`
  See the TypeLoader class Run method.
- (void) - `runTwoPhases`
  See the TypeLoader class RunTwoPhases method.
- (void) - `runNPhases:`
  See the TypeLoader class RunNPhases method.
- (void) - `setLogError:`
  Sets a log error file.
- (void) - `setLogOff`
  Truns off all the error reporting.
- (void) - `registerListener:`
  Registers a new listener.
- (void) - `setRowReader:`
  Sets the input data source.
- (void) - `setGraph:``
  Sets the graph where the data will be loaded.
- (void) - `setLocale:`
  Sets the locale that will be used to read the data.
- (void) - `setType:`
  Sets the type to be loaded.
- (void) - `setAttributes:`
  Sets the list of Attributes.
- (void) - `setAttributePositions:`
  Sets the list of attribute positions.
- (void) - `setTimestampFormat:`
3.30 STSNodeTypeLoader Class Reference

Sets a specific timestamp format.

- (void) - setFrequency:
  Sets the frequency of listener notification.

3.30.1 Detailed Description

NodeTypeLoader class.
Specific TypeLoader implementation for node types.
Check out the 'Data import' section in the SPARKSEE User Manual for more details on this.

Author
Sparsity Technologies http://www.sparsity-technologies.com

3.30.2 Method Documentation

3.30.2.1 - (id) initWithRowReader:  (STSRowReader *) rowReader graph:(STSGraph *) graph type:(int) type attrs:(STSAttributeList *) attrs attrsPos: (STSInt32List *) attrsPos

Creates a new instance.

Parameters

| rowReader | [in] Input RowReader. |
| graph     | [in] Graph. |
| type      | [in] Type identifier. |
| attrs     | [in] Attribute identifiers to be loaded. |
| attrsPos  | [in] Attribute positions (column index >=0). |

3.30.2.2 - (void) registerListener:  (STSTypeLoaderListener *) tel

Registers a new listener.

Parameters

| tel | TypeLoaderListener to be registered. |

3.30.2.3 - (void) run

See the TypeLoader class Run method.

Exceptions

| System.IO.IOException | null |
| System.ApplicationException | null |

Implements STSTypeLoader.

3.30.2.4 - (void) runNPhases:  (int) partitions

See the TypeLoader class RunNPhases method.

Parameters

| partitions | null |
3.30 STSNodeTypeLoader Class Reference

Exceptions

<table>
<thead>
<tr>
<th>System.IO.IOException</th>
<th>null</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.ApplicationException</td>
<td>null</td>
</tr>
</tbody>
</table>

Implements STSTypeLoader.

### 3.30.2.5 - (void) runTwoPhases

See the TypeLoader class RunTwoPhases method.

Exceptions

<table>
<thead>
<tr>
<th>System.IO.IOException</th>
<th>null</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.ApplicationException</td>
<td>null</td>
</tr>
</tbody>
</table>

Implements STSTypeLoader.

### 3.30.2.6 - (void) setAttributePositions: (STSInt32List *) attrsPos

Sets the list of attribute positions.

Parameters

| attrsPos | [in] Attribute positions (column index >=0). |

### 3.30.2.7 - (void) setAttributes: (STSAttributeList *) attrs

Sets the list of Attributes.

Parameters

| attrs | [in] Attribute identifiers to be loaded |

### 3.30.2.8 - (void) setFrequency: (int) freq

Sets the frequency of listener notification.

Parameters

| freq | [in] Frequency in number of rows managed to notify progress to all listeners |

### 3.30.2.9 - (void) setGraph: (STSGraph *) graph

Sets the graph where the data will be loaded.

Parameters

| graph | [in] Graph. |

### 3.30.2.10 - (void) setLocale: (NSString *) localeStr

Sets the locale that will be used to read the data.

It should match the locale used in the rowreader.
3.31 STSObjects Class Reference

Parameters

| localeStr | [in] The locale string for the read data. See CSVReader. |

3.30.2.11 - (void) setError: (NSString *) path

Sets a log error file.

By default errors are thrown as an exception and the load process ends. If a log file is set, errors are logged there and the load process does not stop.

Parameters

| path | [in] The path to the error log file. |

Exceptions

| System.IO.IOException | If bad things happen opening the file. |

3.30.2.12 - (void) setLogOff

TURNS off all the error reporting.

The log file will not be created and no exceptions for invalid data will be thrown. If you just want to turn off the logs, but abort at the first error what you should do is not call this method and not set a logError file.

3.30.2.13 - (void) setRowReader: (STSRowReader *) rr

Sets the input data source.

Parameters

| rr | [in] Input RowReader. |

3.30.2.14 - (void) setTimestampFormat: (NSString *) timestampFormat

Sets a specific timestamp format.

Parameters

| timestampFormat | [in] A string with the timestamp format definition. |

3.30.2.15 - (void) setType: (int) type

Sets the type to be loaded.

Parameters

| type | [in] Type identifier. |

The documentation for this class was generated from the following file:

- Sparksee.h
## Inheritance diagram for STSObjects:

```
  NSObject
     |        |
     |        |
     |        |
  STSObjects
```

## Collaboration diagram for STSObjects:

```
  NSObject
     |        |
     |        |
     |        |
  STSObjects
```

### Instance Methods

- **(STSObjects *) - clone**
  
  Creates a new Objects instance as a copy of the given one.

- **(long long) - count**
  
  Gets the number of elements into the collection.

- **(BOOL) - add:**
  
  Adds an element into the collection.

- **(BOOL) - exists:**
  
  Gets if the given element exists into the collection.

- **(long long) - any**
  
  Gets an element from the collection.

- **(BOOL) - remove:**
  
  Removes an element from the collection.

- **(void) - clear**
  
  Clears the collection removing all its elements.

- **(long long) - union:**
  
  Performs the union operation.

- **(long long) - intersection:**
Perform the intersection operation.
• (long long) - difference:
  Performs the difference operation.
• (BOOL) - equals:
  Checks if the given Objects contains the same information.
• (BOOL) - contains:
  Check if this objects contains the other one.
• (long long) - cloneWithObjects:
  Performs the copy operation.
• (STSObjects ∗) - sample:samples:
  Creates a new Objects instance which is a sample of the calling one.
• (STSObjectsIterator ∗) - iterator
  Gets an ObjectsIterator.
• (STSObjectsIterator ∗) - iteratorFromIndex:
  Gets an ObjectsIterator skipping index elements.
• (STSObjectsIterator ∗) - iteratorFromElement:
  Gets an ObjectsIterator starting from the given element.
• (void) - close
  Closes the Objects instance.
• (BOOL) - isClosed
  Check if the Objects instance is closed.
• (BOOL) - isEqual:
  Check if both Objects instances are equal.
• (NSUInteger) - hash
  Get the hash value of this Objects.

Class Methods
• (long long) + getInvalidOID
  Invalid OID constant.
• (STSObjects ∗) + combineUnion:objs2:
  Creates a new Objects instance which is the union of the two given.
• (STSObjects ∗) + combineIntersection:objs2:
  Creates a new Objects instance which is the intersection of the two given.
• (STSObjects ∗) + combineDifference:objs2:
  Creates a new Objects instance which is the difference of the two given.

3.31.1 Detailed Description

Object identifier set class.
It stores a collection of Sparksee object identifiers as a set. As a set, there is no order and no duplicated elements.
This class should be used just to store large collections. Otherwise, it is strongly recommended to use common
classes from the language API.
This class is not thread-safe.
ObjectsIterator must be used to traverse all the elements into the set.
When the Objects instance is closed, it closes all existing and non-closed ObjectsIterator instances too.

Author
Sparsity Technologies [http://www.sparsity-technologies.com](http://www.sparsity-technologies.com)
3.31 Method Documentation

3.31.2 - (BOOL) add: (long long) e
Adds an element into the collection.

Parameters

| e | [in] Element to be added. |

Returns

TRUE if the element is added, FALSE if the element was already into the collection.

3.31.2.2 - (long long) any
Gets an element from the collection.

Returns

Any element from the collection.

Exceptions

| System.Application-Exception | whether the collection is empty. |
| System.Application-Exception | null |

3.31.2.3 - (STSObjects*) clone
Creates a new Objects instance as a copy of the given one.

Returns

The new Objects instance.

3.31.2.4 - (long long) cloneWithObjects: (STSObjects*) objs
Performs the copy operation.
This updates the Objects calling instance and copies the given Objects instance.

Parameters

| objs | [in] Objects instance. |

Returns

Number of elements into the collection once the operation has been executed.

3.31.2.5 - (void) close
Closes the Objects instance.
It must be called before closing the Session to ensure the integrity of all data.
3.31.2.6 + (STSObjects *) combineDifference: (STSObjects *) objs1 objs2:

Creates a new Objects instance which is the difference of the two given.
Two given Objects belong to the same Session.

Parameters

<table>
<thead>
<tr>
<th>objs1</th>
<th>[in] Objects instance.</th>
</tr>
</thead>
<tbody>
<tr>
<td>objs2</td>
<td>[in] Objects instance.</td>
</tr>
</tbody>
</table>

Returns

New Objects instance.

3.31.2.7 + (STSObjects *) combineIntersection: (STSObjects *) objs1 objs2:

Creates a new Objects instance which is the intersection of the two given.
Two given Objects belong to the same Session.

Parameters

<table>
<thead>
<tr>
<th>objs1</th>
<th>[in] Objects instance.</th>
</tr>
</thead>
<tbody>
<tr>
<td>objs2</td>
<td>[in] Objects instance.</td>
</tr>
</tbody>
</table>

Returns

New Objects instance.

3.31.2.8 + (STSObjects *) combineUnion: (STSObjects *) objs1 objs2:

Creates a new Objects instance which is the union of the two given.
Two given Objects belong to the same Session.

Parameters

<table>
<thead>
<tr>
<th>objs1</th>
<th>[in] Objects instance.</th>
</tr>
</thead>
<tbody>
<tr>
<td>objs2</td>
<td>[in] Objects instance.</td>
</tr>
</tbody>
</table>

Returns

New Objects instance.

3.31.2.9 -(BOOL) contains: (STSObjects *) objs

Check if this objects contains the other one.

Parameters

| objs  | Objects collection. |
3.31 STSObjects Class Reference

Returns
True if it contains the given object.

3.31.2.10 - (long long) count

Gets the number of elements into the collection.

Returns
The number of elements into the collection.

3.31.2.11 - (long long) difference: (STSObjects * ) objs

Performs the difference operation.
This updates the Objects calling instance removing those existing elements at the given Objects instance.

Parameters

| objs | [in] Objects instance. |

Returns
Number of elements into the collection once the operation has been executed.

3.31.2.12 - (BOOL) equals: (STSObjects * ) objs

Checks if the given Objects contains the same information.

Parameters

| objs | [in] Objects instance. |

Returns
True if the objects are equal or false otherwise.

3.31.2.13 - (BOOL) exists: (long long) e

Gets if the given element exists into the collection.

Parameters

| e | [in] Element. |

Returns
TRUE if the element exists into the collection, FALSE otherwise.

3.31.2.14 - (long long) intersection: (STSObjects * ) objs

Performs the intersection operation.
Updates the Objects calling instance setting those existing elements at both two collections and removing all others.

Parameters

| objs | [in] Objects instance. |
Returns

Number of elements into the collection once the operation has been executed.

3.31.2.15 - (STSObjectsIterator *) iterator

Gets an ObjectsIterator.

Returns

ObjectsIterator instance.

3.31.2.16 - (STSObjectsIterator *) iteratorFromElement: (long long) e

Gets an ObjectsIterator starting from the given element.

Objects collection has no order, so this method is implementation-dependent. e[in] The first element to traverse in the resulting

Parameters

| e   | [in] The first element to traverse in the resulting ObjectsIterator instance. |

Returns

ObjectsIterator instance.

3.31.2.17 - (STSObjectsIterator *) iteratorFromIndex: (long long) index

Gets an ObjectsIterator skipping index elements.

Objects collection has no order, so this method is implementation-dependent.

Parameters

| index | [in] The number of elements to skip from the beginning. It must be in the range [0..Size). |

Returns

ObjectsIterator instance.

3.31.2.18 - (BOOL) remove: (long long) e

Removes an element from the collection.

Parameters

| e   | [in] Element to be removed. |

Returns

TRUE if the element is removed, FALSE if the element was not into the collection.

3.31.2.19 - (STSObjects *) sample: (STSObjects *) exclude samples:(long long) samples

Creates a new Objects instance which is a sample of the calling one.
Parameters

| exclude | [in] If not NULL, elements into this collection will be excluded from the resulting one. |
| samples | [in] Number of elements into the resulting collection. |

Returns

Sample collection.

3.31.2.20 - (long long) union: (STSObjects *) objs

Performs the union operation.
This adds all existing elements of the given Objects instance to the Objects calling instance

Parameters

| objs | [in] Objects instance. |

Returns

Number of elements into the collection once the operation has been executed.

The documentation for this class was generated from the following file:

- Sparksee.h

3.32 STSObjectsIterator Class Reference

ObjectsIterator class.

Inheritance diagram for STSObjectsIterator:
Collaboration diagram for STSObjectsIterator:

![Collaboration Diagram](https://example.com/collaboration-diagram.png)

**Instance Methods**

- (BOOL) - hasNext
  
  Gets if there are more elements to traverse.

- (long long) - next
  
  Gets the next element to traverse.

- (void) - close
  
  Closes the ObjectsIterator instance.

- (BOOL) - isClosed
  
  Check if the ObjectsIterator instance is closed.

### 3.32.1 Detailed Description

ObjectIterator class.

Iterator to traverse all the object identifiers from an Objects instance.

**Author**

Sparsity Technologies [http://www.sparsity-technologies.com](http://www.sparsity-technologies.com)

### 3.32.2 Method Documentation

#### 3.32.2.1 - (void) close

Closes the ObjectIterator instance.

It must be called before closing the parent Objects.

#### 3.32.2.2 - (BOOL) hasNext

Gets if there are more elements to traverse.

Returns

TRUE if there are more elements to traverse, FALSE otherwise.
3.32.2.3  -(long long) next

Gets the next element to traverse.

Returns
   The next element.

The documentation for this class was generated from the following file:

   • Sparksee.h

3.33  STS OID List Class Reference

Sparksee object identifier list.

Inheritance diagram for STS OID List:

![Inheritance diagram for STS OID List]

Collaboration diagram for STS OID List:

![Collaboration diagram for STS OID List]

Instance Methods

- (int) - count
  
  Number of elements in the list.

- (id) - init
Constructor.

- (id) - initWithNumInvalidOIDs:
  Constructor.
- (void) - add:
  Adds a Sparksee object identifier at the end of the list.
- (void) - set:oid:
  Sets a Sparksee object identifier at the specified position of the list.
- (void) - clear
  Clears the list.
- (id) - initWithArray:
  Creates a new OIDList instance from the given array.
- (id) - initWithNSEnumerator:
  Creates a new OIDList instance from the given NSEnumerator.
- (STSOidListIterator *) - iterator
  Gets a new OIDListIterator.

3.33.1 Detailed Description

Sparksee object identifier list.
It stores a Sparksee object identifier list.
Use OIDListIterator to access all elements into this collection.

Author

Sparsity Technologies [http://www.sparsity-technologies.com](http://www.sparsity-technologies.com)

3.33.2 Method Documentation

3.33.2.1 - (void) add: (long long) attr

Adds a Sparksee object identifier at the end of the list.

Parameters

| attr   | [in] Sparksee object identifier. |

3.33.2.2 - (int) count

Number of elements in the list.

Returns

Number of elements in the list.

3.33.2.3 - (id) init

Constructor.
This creates an empty list.

3.33.2.4 - (id) initWithNumInvalidOIDs: (int) numInvalidOIDs

Constructor.
This creates a list with N invalid oids.
Parameters

\[ \text{numInvalidOIDs} \quad \text{[in]} \quad \text{The number of invalid oids added to the list.} \]

3.33.2.5 \quad \text{- (void) set: (int) pos:(long long) oid}

Sets a Sparksee object identifier at the specified position of the list.

Parameters

<table>
<thead>
<tr>
<th>pos</th>
<th>[in] List position [0..Count()-1].</th>
</tr>
</thead>
<tbody>
<tr>
<td>oid</td>
<td>[in] Sparksee object identifier.</td>
</tr>
</tbody>
</table>

The documentation for this class was generated from the following file:

- Sparksee.h

3.34 \quad \text{STSOidListIterator Class Reference}

OIDList iterator class.

Inheritance diagram for STSOidListIterator:
Collaboration diagram for STSOidListIterator:

```
NSObject

STSOidList

theParent

STSOidListIterator
```

Instance Methods

- (long long) - next
  
  Moves to the next element.

- (BOOL) - hasNext
  
  Gets if there are more elements.

3.34.1 Detailed Description

OIDList iterator class.
Iterator to traverse all the Sparksee object identifier into a OIDList instance.

Author

Sparsity Technologies [http://www.sparsity-technologies.com](http://www.sparsity-technologies.com)

3.34.2 Method Documentation

3.34.2.1 - (BOOL) hasNext

Gets if there are more elements.

Returns

TRUE if there are more elements, FALSE otherwise.

3.34.2.2 - (long long) next

Moves to the next element.
Returns

The next element.

The documentation for this class was generated from the following file:

- Sparksee.h

### 3.35 STSPlatform Class Reference

Platform class.

Inheritance diagram for STSPlatform:

```
 NSObject
     |
     V
STSPlatform
```

Collaboration diagram for STSPlatform:

```
 NSObject
     |
     V
STSPlatform
```

### Class Methods

- (void) + **getStatistics**:

  *Gets platform data and statistics.*

### 3.35.1 Detailed Description

Platform class.
3.35.2 Method Documentation

3.35.2.1 + (void) getStatistics: (STSPlatformStatistics *) stats

Gets platform data and statistics.

Parameters

| stats | [in|out] This updates the given PlatformStatistics. |

The documentation for this class was generated from the following file:

- Sparksee.h

3.36 STSPlatformStatistics Class Reference

Platform data and statistics.

Inheritance diagram for STSPlatformStatistics:

```
  NSObject
    
  STSPlatformStatistics
```

Collaboration diagram for STSPlatformStatistics:

```
  NSObject
    
  STSPlatformStatistics
```

Author

Sparsity Technologies [http://www.sparsity-technologies.com](http://www.sparsity-technologies.com)
3.36.1 Detailed Description

Platform data and statistics.

Author

Sparsity Technologies [http://www.sparsity-technologies.com](http://www.sparsity-technologies.com)

3.36.2 Method Documentation

3.36.2.1 - (long long) getAvailableMem

Gets available (free) memory size in Bytes.

Returns

Avialable (free) memory size in Bytes.

3.36.2.2 - (int) getNumCPUs

Gets the number of CPUs.

Returns

The number of CPUs.

3.36.2.3 - (long long) getRealTime

Gets time in microseconds (since epoch).

Returns

Time in microseconds (since epoch).

3.36.2.4 - (long long) getSystemTime

Gets CPU system time.

Returns

CPU system time.
3.36.2.5 - (long long) getTotalMem

Gets physical memory size in Bytes.

Returns

Physical memory size in Bytes.

3.36.2.6 - (long long) getUserTime

Gets CPU user time.

Returns

CPU user time.

The documentation for this class was generated from the following file:

• Sparksee.h

3.37 STSQuery Class Reference

Query class.

Inheritance diagram for STSQuery:

```
  NSObject
   ^
  STSQuery
```

Collaboration diagram for STSQuery:

```
  NSObject
   ^
  STSQuery
```
Instance Methods

- (STSResultSet *) execute:
  Executes the given statement.
- (STSQueryStream *) setStream:handler:
  Sets a query stream handler.
- (void) setDynamic:value:
  Sets the value for a dynamic parameter.
- (void) close
  Closes the Query instance.
- (BOOL) isClosed
  Check if the Query instance is closed.

3.37.1 Detailed Description

Query class.

Author

Sparsity Technologies [http://www.sparsity-technologies.com](http://www.sparsity-technologies.com)

3.37.2 Method Documentation

3.37.2.1 - (STSResultSet *) execute: (NSString *) stmt

Executes the given statement.

Parameters

<table>
<thead>
<tr>
<th>name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stmt</td>
<td>Query statement.</td>
</tr>
</tbody>
</table>

Returns

A ResultSet instance with the contents of the result of the query.

3.37.2.2 - (void) setDynamic: (NSString *) name value:(STSValue *) value

Sets the value for a dynamic parameter.

Parameters

<table>
<thead>
<tr>
<th>name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>Parameter value</td>
</tr>
</tbody>
</table>

3.37.2.3 - (STSQueryStream *) setStream: (NSString *) stream handler:(STSQueryStream *) handler

Sets a query stream handler.

Query streams handlers are created and destroyed by the caller.

Parameters

<table>
<thead>
<tr>
<th>name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stream</td>
<td>The stream name</td>
</tr>
<tr>
<td>handler</td>
<td>Query stream handler</td>
</tr>
</tbody>
</table>
Returns

The previous handler, or NULL if it does not exists

The documentation for this class was generated from the following file:

- Sparksee.h

3.38 STSQueryContext Class Reference

Query context interface.

Inheritance diagram for STSQueryContext:

```
NSObject
    ^
STSQueryContext
```

Collaboration diagram for STSQueryContext:

```
NSObject
    ^
STSQueryContext
```

Instance Methods

- (id) - init
  Default constructor.
- (STSQuery *) - createQuery
  Creates a new Query.
3.38.1 Detailed Description

Query context interface.

A QueryContext contains and manages the resources required to run a Query. A Session is one example of a QueryContext connected to a Sparksee database. The applications can implement their own contexts to run queries out of Sparksee.

Author

Sparsity Technologies [http://www.sparsity-technologies.com](http://www.sparsity-technologies.com)

The documentation for this class was generated from the following file:

- Sparksee.h

3.39 STSQueryStream Class Reference

Query stream interface.

Inheritance diagram for STSQueryStream:

![Inheritance Diagram](image)

Collaboration diagram for STSQueryStream:

![Collaboration Diagram](image)

Instance Methods

- (BOOL) prepare:
Prepares the stream before it is started.

- (BOOL) - start:
  Starts the stream.
- (BOOL) - fetch:
  Gets the next row and moves the iterator forward.

### 3.39.1 Detailed Description

Query stream interface.

A QueryStream is the interface between the application and the STREAM operator. When the operator starts inside a Query, the method is prepared with query-defined arguments. Then, if there are input operations, the STREAM operator builds the ResultSets and starts the iteration. Finally, the operator fetches rows until no more are available.

Application exceptions must be cached by the subclass that implements the interface.

**Author**

Sparsity Technologies [http://www.sparsity-technologies.com](http://www.sparsity-technologies.com)

### 3.39.2 Method Documentation

#### 3.39.2.1 - (BOOL) fetch: (STSValueList *) list

Gets the next row and moves the iterator forward.

The end of sequence is denoted by returning TRUE with an empty row. A valid row must contain as many values (even NULL) as expected by the query.

**Parameters**

| list | [out] Storage for the new rows |

**Returns**

TRUE if there is a row or end of sequence, FALSE on error

#### 3.39.2.2 - (BOOL) prepare: (STSValueList *) list

Prepares the stream before it is started.

**Parameters**

| list | [in] Optional list of arguments |

**Returns**

FALSE on error

#### 3.39.2.3 - (BOOL) start: (STSResultSetList *) list

Starts the stream.

**Parameters**

| list | [in] Optional list of input ResultSets |
Returns

FALSE on error

The documentation for this class was generated from the following file:

- Sparksee.h

### 3.40 STSResultSet Class Reference

ResultSet class.

Inheritance diagram for STSResultSet:

![Inheritance Diagram](image)

Collaboration diagram for STSResultSet:

![Collaboration Diagram](image)

### Instance Methods

- (int) - `getNumColumns`
  
  *Gets the number of columns.*

- (NSString *) - `getColumnName`
  
  *Gets the name for the given column.*

- (int) - `getColumnIndex`
  
  *Gets the column index for the given column name.*

- (enum STSDataType) - `getColumnDataType`
  
  *Gets the column data type for the given column name.*
3.40 STSResultSet Class Reference

3.40.1 Detailed Description

ResultSet class.

Author

Sparsity Technologies http://www.sparsity-technologies.com

3.40.2 Method Documentation

3.40.2.1 - (STSValue *) getColumn: (int) index

Gets the value for the given column.

QueryException If a database access error occurs.

Parameters

| index | in Column index. |

Returns

The Value of the given column.

3.40.2.2 - (enum STSDataType) getColumnDataType: (int) index

Gets the datatype for the given column.

Parameters

| index | in Column index. |

Returns

DataType for the given column.

3.40.2.3 - (int) getColumnIndex: (NSString *) name

Gets the column index for the given column name.

Generated on Tue Jun 2 2015 15:49:36 for Sparksee by Doxygen
Parameters

| name | [in] Column name. |

Returns

Column index.

3.40.2.4 - (NSString *) getColumnName: (int) index

Gets the name for the given column.

Parameters

| index | [in] Column index. |

Returns

Column name.

3.40.2.5 - (void) getColumnWithValue: (int) index:(STSValue *) value

Gets the value for the given column.

QueryExceptionIf a database access error occurs.

Parameters

| index | [in] Column index. |
| value | [in/out] Value. |

3.40.2.6 - (NSString *) getJSON: (int) rows

Returns rows in JSON format.

Rows are returned from the current position.

Parameters

| rows | [in] Maximum number of rows |

Returns

JSON representation of the next <rows> rows in the resultset.

3.40.2.7 - (int) getNumColumns

Gets the number of columns.

Columns are in the range [0...COLUMNS).

Returns

The number of columns.

3.40.2.8 - (BOOL) next

Fetches the next row.

A ResultSet cursor is initially positioned before the first row; the first call to the method "Next" makes the first row the current row; the second call makes the second row the current row, and so on.
QueryExceptionIf a database access error occurs.

Returns
   TRUE if the next row has been successfully fetched, FALSE otherwise.

The documentation for this class was generated from the following file:
   • Sparksee.h

3.41 STSResultSetList Class Reference

ResultSet list.

Inheritance diagram for STSResultSetList:

Collaboration diagram for STSResultSetList:

Instance Methods
   • (int) - count
      Number of elements in the list.
   • (id) - init
      Constructor.
   • (void) - clear
3.42 STSResultSetListIterator Class Reference

Clears the list.

- (STSResultSet *) - get:
  Returns the ResultSet at the specified position in the list.

- (STSResultSetListIterator *) - iterator
  Gets a new ResultSetListIterator.

3.41 Detailed Description

ResultSet list.
It stores a ResultSet list.

Author
Sparsity Technologies http://www.sparsity-technologies.com

3.41.2 Method Documentation

3.41.2.1 - (int) count
Number of elements in the list.

Returns
Number of elements in the list.

3.41.2.2 - (STSResultSet*) get: (int) index
Returns the ResultSet at the specified position in the list.

Parameters

| index | [in] Index of the element to return, starting at 0. |

3.41.2.3 - (id) init
Constructor.
This creates an empty list.
The documentation for this class was generated from the following file:

- Sparksee.h

3.42 STSResultSetListIterator Class Reference

ResultSetList iterator class.
Inheritance diagram for STSResultSetListIterator:

Collaboration diagram for STSResultSetListIterator:

Instance Methods

- (STSResultSet *) - next
  
  Moves to the next element.
- (BOOL) - hasNext
  
  Gets if there are more elements.

3.42.1 Detailed Description

ResultSetList iterator class.

Iterator to traverse all the values into a ResultSetList instance.
Author
Sparsity Technologies http://www.sparsity-technologies.com

3.42.2 Method Documentation

3.42.2.1 - (BOOL) hasNext

Gets if there are more elements.

Returns
TRUE if there are more elements, FALSE otherwise.

3.42.2.2 - (STSCSVReader *) next

Moves to the next element.

Returns
The next element.

The documentation for this class was generated from the following file:

• Sparksee.h

3.43 STSRowReader Class Reference

RowReader interface.
Inheritance diagram for STSRowReader:

```
            NSObject
               ^
              |   
            STSRowReader
               |   
          STSCSVReader
```

Generated on Tue Jun 2 2015 15:49:36 for Sparksee by Doxygen
Collaboration diagram for STSRowReader:

```
NSObject

STSRowReader
```

Instance Methods

- (BOOL) - reset
  Moves the reader to the beginning.
- (BOOL) - read:
  Reads the next row as a string array.
- (int) -getRow
  The row number for the current row.
- (void) - close
  Closes the reader.

3.43.1 Detailed Description

RowReader interface.

Common interface for those readers which get the data as a string array.

It works as follows: perform as many read operations as necessary and call close once at the end. Once close is called no more read operations can be executed.

Check out the ‘Data import’ section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies [http://www.sparsity-technologies.com](http://www.sparsity-technologies.com)

3.43.2 Method Documentation

3.43.2.1 - (void) close

Closes the reader.

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.IO.IOException</td>
<td>If the close fails.</td>
</tr>
</tbody>
</table>

Implemented in STSCSVReader.
3.43.2.2 - (int)getRow

The row number for the current row.

Returns

The current row number; 0 if there is no current row.

Exceptions

| System.IO.IOException | If it fails. |

Implemented in STSCSVReader.

3.43.2.3 - (BOOL)read: (STSStringList ∗) row

Reads the next row as a string array.

Parameters

 row ∗ [out] A string list with each comma-separated element as a separate entry.

Returns

Returns true if a row had been read or false otherwise.

Exceptions

| System.IO.IOException | If bad things happen during the read. |

Implemented in STSCSVReader.

3.43.2.4 - (BOOL)reset

Moves the reader to the beginning.

Restarts the reader.

Returns

true if the reader can be restarted, false otherwise.

Exceptions

| System.IO.IOException | If bad things happen during the restart. |

Implemented in STSCSVReader.

The documentation for this class was generated from the following file:

- Sparksee.h

3.44 STSRowWriter Class Reference

RowWriter interface.
Inheritance diagram for STSRowWriter:

```
NSObject
  ↓
STSRowWriter
  ↓
STSCSVWriter
```

Collaboration diagram for STSRowWriter:

```
NSObject
  ↓
STSRowWriter
```

Instance Methods

- (void) - write:
  
  * `Write the next row.`

- (void) - close
  
  * `Closes the writer.`

3.44.1 Detailed Description

RowWriter interface.

Common interface for those writers which dump the data from an string array.

It works as follows: perform as many write operations as necessary and call close once at the end. Once close is called no more write operations can be executed.

Check out the 'Data export' section in the SPARKSEE User Manual for more details on this.
3.44.2 Method Documentation

3.44.2.1 - (void) close

Closes the writer.

<table>
<thead>
<tr>
<th>Exceptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.IO.IOException If the close fails.</td>
</tr>
<tr>
<td>System.ApplicationException null</td>
</tr>
</tbody>
</table>

Implemented in STSCSVWriter.

3.44.2.2 - (void) write: (STSStringList *) row

Writes the next row.

<table>
<thead>
<tr>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>row [in] Row of data.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Exceptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.IO.IOException If bad things happen during the write.</td>
</tr>
<tr>
<td>System.ApplicationException null</td>
</tr>
</tbody>
</table>

Implemented in STSCSVWriter.

The documentation for this class was generated from the following file:

- Sparksee.h

3.45 STSScriptParser Class Reference

ScriptParser.

Inheritance diagram for STSScriptParser:
Collaboration diagram for STSScriptParser:

```
  NSObject
   ^
   |  
STSScriptParser
```

Instance Methods

- (id) - init
  Constructor.

- (void) - setOutputLog:
  Sets the output log.

- (void) - setErrorLog:
  Sets the error log.

- (BOOL) - parse:execute:localeStr:
  Parses the given input file.

Class Methods

- (void) + generateSchemaScript:db:
  Writes an script with the schema definition for the given database.

3.45.1 Detailed Description

ScriptParser.

The ScriptParser can create schemas and load data from a set of commands in a sparksee script.

A SPARKSEE script contains an ordered list of commands. ScriptParser will execute each one of them in order. Commands may create schemas, define nodes and edges, and load data into a previous defined SPARKSEE schema.

Check out the ‘Scripting’ chapter in the SPARKSEE User Manual for a comprehensive explanation on the grammar of the SPARKSEE commands and how they work.

Author

Sparsity Technologies [http://www.sparsity-technologies.com](http://www.sparsity-technologies.com)

3.45.2 Method Documentation

3.45.2.1 + (void) generateSchemaScript: (NSString *) path db:(STSDatabase *) db

Writes an script with the schema definition for the given database.
Parameters

<table>
<thead>
<tr>
<th>path</th>
<th>[in] Filename of the script to be written.</th>
</tr>
</thead>
<tbody>
<tr>
<td>db</td>
<td>[in] Database.</td>
</tr>
</tbody>
</table>

Exceptions

| System.IO.IOException | If bad things happen opening or writing the file. |

3.45.2.2 - (BOOL) parse: (NSString *) path execute:(BOOL) execute localeStr:(NSString *) localeStr

 Parses the given input file.

Parameters

<table>
<thead>
<tr>
<th>path</th>
<th>[in] Input file path.</th>
</tr>
</thead>
<tbody>
<tr>
<td>execute</td>
<td>[in] If TRUE the script is executed, if FALSE it is just parsed.</td>
</tr>
<tr>
<td>localeStr</td>
<td>[in] The locale string for reading the input file. See CSVReader.</td>
</tr>
</tbody>
</table>

Returns

TRUE if ok, FALSE in case of error.

Exceptions

| System.IO.IOException | If bad things happen opening the file. |

3.45.2.3 - (void) setErrorLog: (NSString *) path

Sets the error log.
If not set, error log corresponds to standard error output.

Parameters

| path | [in] Path of the error log. |

Exceptions

| System.IO.IOException | If bad things happen opening the file. |

3.45.2.4 - (void) setOutputLog: (NSString *) path

Sets the output log.
If not set, output log corresponds to standard output.

Parameters

| path  | [in] Path of the output log. |

Exceptions

| System.IO.IOException | If bad things happen opening the file. |

The documentation for this class was generated from the following file:

- Sparksee.h
3.46 STSSession Class Reference

Session class.

Inheritance diagram for STSSession:

Collaboration diagram for STSSession:

Instance Methods

- (STSGraph *) - getGraph
  
  Gets the Graph instance.

- (STSObjects *) - createObjects
  
  Creates a new Objects instance.

- (void) - begin
  
  Begins a transaction.

- (void) - beginUpdate
  
  Begins an update transaction.

- (void) - commit
  
  Commits a transaction.

- (void) - rollback
  
  Rollbacks a transaction.

- (STSQuery *) - createQuery
  
  Creates a new Query.
3.47 STSShortestPath Class Reference

ShortestPath class.

3.46  Detailed Description

Session class.

A Session is a stateful period of activity of a user with the Database.

All the manipulation of a Database must be enclosed into a Session. A Session can be initiated from a Database
instance and allows for getting a Graph instance which represents the persistent graph (the graph database).

Also, temporary data is associated to the Session, thus when a Session is closed, all the temporary data associated
to the Session is removed too. Objects or Values instances or even session attributes are an example of temporary
data.

Moreover, a Session is exclusive for a thread, thus if it is shared among threads results may be fatal or unexpected.

Check out the ‘Processing’ and ‘Transactions’ sections in the SPARKSEE User Manual for details about how Ses-
sions work and the use of transactions.

Author

Sparsity Technologies http://www.sparsity-technologies.com

3.46.2  Method Documentation

3.46.2.1  -(void) close

Closes the Session instance.

It must be called to ensure the integrity of all data.

3.46.2.2  -(STSOBjects *) createObjects

Creates a new Objects instance.

Returns

The new Objects instance.

3.46.2.3  -(STSGraph *) getGraph

Gets the Graph instance.

Returns

The Graph instance.

The documentation for this class was generated from the following file:

• Sparksee.h
Inheritance diagram for STSShortestPath:

Collaboration diagram for STSShortestPath:

Instance Methods

- (void) - **setMaximumHops:**
  
  *Sets the maximum hops restriction.*

- (void) - **addEdgeType:**dir:
  
  *Allows for traversing edges of the given type.*

- (void) - **addAllEdgeTypes:**
  
  *Allows for traversing all edge types of the graph.*

- (void) - **addNodeType:**
  
  *Allows for traversing nodes of the given type.*

- (void) - **addAllNodeTypes**
  
  *Allows for traversing all node types of the graph.*
3.47 STSShortestPath Class Reference

• (void) - **excludeNodes**: 
  Set which nodes can’t be used.
• (void) - **excludeEdges**: 
  Set which edges can’t be used.
• (void) - **run**
  Runs the algorithm.
• (void) - **close**
  Closes the ShortestPath instance.
• (BOOL) - **isClosed**
  Check if the ShortestPath instance is closed.

3.47.1 Detailed Description

ShortestPath class.

Classes implementing this abstract class solve the shortest path problem in a graph.
The user must set which node and edge types can be used for the traversal.
Check out the 'Algorithms' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies [http://www.sparsity-technologies.com](http://www.sparsity-technologies.com)

3.47.2 Method Documentation

3.47.2.1 - (void) addAllEdgeTypes: (enum STSEdgesDirection) **dir**

Allows for traversing all edge types of the graph.

Parameters

| dir  | [in] Edge direction. |

3.47.2.2 - (void) addEdgeType: (int) **type**:(enum STSEdgesDirection) **dir**

Allows for traversing edges of the given type.

Parameters

| type | [in] Edge type. |
| dir  | [in] Edge direction. |

3.47.2.3 - (void) addNodeType: (int) **type**

Allows for traversing nodes of the given type.

Parameters

| type | null |

3.47.2.4 - (void) close

Closes the ShortestPath instance.
It must be called to ensure the integrity of all data.
3.47.2.5 - (void) excludeEdges: (STSObjects *) edges

Set which edges can't be used.
This will replace any previously specified set of excluded edges. Should only be used to exclude the usage of specific edges from allowed edge types because it's less efficient than not allowing an edge type.

Parameters

| edges | [in] A set of edge identifiers that must be kept intact until the destruction of the class. |

3.47.2.6 - (void) excludeNodes: (STSObjects *) nodes

Set which nodes can't be used.
This will replace any previously specified set of excluded nodes. Should only be used to exclude the usage of specific nodes from allowed node types because it's less efficient than not allowing a node type.

Parameters

| nodes | [in] A set of node identifiers that must be kept intact until the destruction of the class. |

3.47.2.7 - (void) run

Runs the algorithm.
This method can only be called once.
Implemented in STSSinglePairShortestPathDijkstra, and STSSinglePairShortestPathBFS.

3.47.2.8 - (void) setMaximumHops: (int) maxhops

Sets the maximum hops restriction.
All paths longer than the maximum hops restriction will be ignored.

Parameters

| maxhops | [in] The maximum hops restriction. It must be positive or zero. Zero, the default value, means unlimited. |

The documentation for this class was generated from the following file:

- Sparksee.h

3.48 STSSinglePairShortestPath Class Reference

SinglePairShortestPath class.
Inheritance diagram for STSSinglePairShortestPath:

```
NSObject
   ↓
STSShortestPath
   ↓
STSSinglePairShortestPath
   ↓
STSSinglePairShortestPathBFS  STSSinglePairShortestPathDijkstra
```

Collaboration diagram for STSSinglePairShortestPath:

```
NSObject
   ↓
STSShortestPath
   ↓
STSSinglePairShortestPath
```

Instance Methods

- `(STSOidList *) - getPathAsNodes
  Gets the shortest path between the source node and the destination node as an ordered set of nodes.
- `(STSOidList *) - getPathAsEdges
  Gets the shortest path between the source node and the destination node as an ordered set of edges.
- `(double) - getCost
  Gets the cost of the shortest path.
3.48 STSSinglePairShortestPath Class Reference

- (BOOL) - exists
  Returns TRUE if a path exists or FALSE otherwise.

- (void) - setMaximumHops:
  Sets the maximum hops restriction.

- (void) - addEdgeType:dir:
  Allows for traversing edges of the given type.

- (void) - addAllEdgeTypes:
  Allows for traversing all edge types of the graph.

- (void) - addNodeType:
  Allows for traversing nodes of the given type.

- (void) - addAllNodeTypes
  Allows for traversing all node types of the graph.

- (void) - excludeNodes:
  Set which nodes can't be used.

- (void) - excludeEdges:
  Set which edges can't be used.

- (void) - run
  Runs the algorithm.

- (void) - close
  Closes the ShortestPath instance.

- (BOOL) - isClosed
  Check if the ShortestPath instance is closed.

3.48.1 Detailed Description

SinglePairShortestPath class.

Classes implementing this abstract class solve the shortest path problem in a graph from a given source node and to a given destination node.

Check out the 'Algorithms' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com

3.48.2 Method Documentation

3.48.2.1 - (void) addAllEdgeTypes: (enum STSEdgesDirection) dir

Allows for traversing all edge types of the graph.

Parameters

<table>
<thead>
<tr>
<th>dir</th>
<th>[in] Edge direction.</th>
</tr>
</thead>
</table>

3.48.2.2 - (void) addEdgeType: (int) type:(enum STSEdgesDirection) dir

Allows for traversing edges of the given type.

Parameters

<table>
<thead>
<tr>
<th>type</th>
<th>[in] Edge type.</th>
</tr>
</thead>
<tbody>
<tr>
<td>dir</td>
<td>[in] Edge direction.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
3.48.2.3 - (void) addNodeType: (int) type

Allows for traversing nodes of the given type.

Parameters

| type | null |

3.48.2.4 - (void) close

Closes the ShortestPath instance.

It must be called to ensure the integrity of all data.

3.48.2.5 - (void) excludeEdges: (STSObjects *) edges

Set which edges can’t be used.

This will replace any previously specified set of excluded edges. Should only be used to exclude the usage of specific edges from allowed edge types because it’s less efficient than not allowing an edge type.

Parameters

| edges | [in] A set of edge identifiers that must be kept intact until the destruction of the class. |

3.48.2.6 - (void) excludeNodes: (STSObject *) nodes

Set which nodes can’t be used.

This will replace any previously specified set of excluded nodes. Should only be used to exclude the usage of specific nodes from allowed node types because it’s less efficient than not allowing a node type.

Parameters

| nodes | [in] A set of node identifiers that must be kept intact until the destruction of the class. |

3.48.2.7 - (double) getCost

Gets the cost of the shortest path.

The cost for unweighted algorithms is the number of hops of the shortest path. For weighted algorithms, the cost is the sum of the costs of the edges of the shortest path.

Returns

The cost of the shortest path.

Implemented in STSSinglePairShortestPathDijkstra, and STSSinglePairShortestPathBFS.

3.48.2.8 - (STSOidList *) getPathAsEdges

Gets the shortest path between the source node and the destination node as an ordered set of edges.

Returns

Ordered set of edge identifiers.

Implemented in STSSinglePairShortestPathDijkstra, and STSSinglePairShortestPathBFS.

3.48.2.9 - (STSOidList *) getPathAsNodes

Gets the shortest path between the source node and the destination node as an ordered set of nodes.

Generated on Tue Jun 2 2015 15:49:36 for Sparksee by Doxygen
Returns
Ordered set of node identifiers.

Implemented in STSSinglePairShortestPathDijkstra, and STSSinglePairShortestPathBFS.

3.48.2.10 - (void) run
Runs the algorithm.
This method can only be called once.
Implemented in STSSinglePairShortestPathDijkstra, and STSSinglePairShortestPathBFS.

3.48.2.11 - (void) setMaximumHops: (int) maxhops
Sets the maximum hops restriction.
All paths longer than the maximum hops restriction will be ignored.

Parameters

| maxhops   | [in] The maximum hops restriction. It must be positive or zero. Zero, the default value, means unlimited. |

The documentation for this class was generated from the following file:

- Sparksee.h

3.49 STSSinglePairShortestPathBFS Class Reference

SinglePairShortestPathBFS class.
Inheritance diagram for STSSinglePairShortestPathBFS:
Collaboration diagram for STSSinglePairShortestPathBFS:

```
+----------------+            +----------------+            +----------------+            +----------------+            +----------------+
|                |                        |                |                        |                |                        |
| STSSinglePairShortestPathBFS |                  | STSSinglePairShortestPath |                  | STSSinglePairShortestPath |                  |
|                |                        |                |                        |                |                        |
| STSSinglePairShortestPath |                  | STSShortestPath |                        | STSSinglePairShortestPath |                  |
|                |                        |                |                        |                |                        |
| STSShortestPath |                        |                |                        |                |                        |
|                |                        |                |                        |                |                        |
| NSObject |                        |                |                        |                |                        |
```

Instance Methods

- (void) - run
  
  Executes the algorithm.

- (STSOidList *) - getPathAsNodes
  
  Gets the shortest path between the source node and the destination node as an ordered set of nodes.

- (STSOidList *) - getPathAsEdges
  
  Gets the shortest path between the source node and the destination node as an ordered set of edges.

- (double) - getCost
  
  Gets the cost of the shortest path.

- (id) - initWithSession:src:dst:
  
  Creates a new instance.

- (void) - checkOnlyExistence
  
  Set that only the path existence must be calculated and not the path itself.

- (BOOL) - exists
  
  Returns TRUE if a path exists or FALSE otherwise.

- (void) - setMaximumHops:
  
  Sets the maximum hops restriction.

- (void) - addEdgeType:dir:
  
  Allows for traversing edges of the given type.

- (void) - addAllEdgeTypes:
  
  Allows for traversing all edge types of the graph.

- (void) - addNodeType:
  
  Allows for traversing nodes of the given type.

- (void) - addAllNodeTypes
  
  Allows for traversing all node types of the graph.
3.49 STSSinglePairShortestPathBFS Class Reference

- (void) - **excludeNodes**:  
  *Set which nodes can’t be used.*
- (void) - **excludeEdges**:  
  *Set which edges can’t be used.*
- (void) - **close**  
  *Closes the ShortestPath instance.*
- (BOOL) - **isClosed**  
  *Check if the ShortestPath instance is closed.*

3.49.1 Detailed Description

**SinglePairShortestPathBFS class.**

It solves the single-pair shortest path problem using a BFS-based implementation.  
It is a unweighted algorithm, that is it assumes all edges have the same cost.  
Check out the ‘Algorithms’ section in the SPARKSEE User Manual for more details on this.

**Author**

Sparsity Technologies [http://www.sparsity-technologies.com](http://www.sparsity-technologies.com)

3.49.2 Method Documentation

3.49.2.1 - (void) addAllEdgeTypes: (enum STSEdgesDirection) **dir**

Allows for traversing all edge types of the graph.

**Parameters**

```
dir | [in] Edge direction.
```

3.49.2.2 - (void) addEdgeType: (int) **type**: (enum STSEdgesDirection) **dir**

Allows for traversing edges of the given type.

**Parameters**

```
type | [in] Edge type.  
dir  | [in] Edge direction.
```

3.49.2.3 - (void) addNodeType: (int) **type**

Allows for traversing nodes of the given type.

**Parameters**

```
type | null
```

3.49.2.4 - (void) checkOnlyExistence

Set that only the path existence must be calculated and not the path itself.  
That method should improve the performance of the algorithm, but a call to GetPathAsNodes or GetPathAsEdges will generate an exception even if the path exists.
3.49.2.5 - (void) close

Closes the ShortestPath instance.
It must be called to ensure the integrity of all data.

3.49.2.6 - (void) excludeEdges: (STSObjects *) edges

Set which edges can’t be used.
This will replace any previously specified set of excluded edges. Should only be used to exclude the usage of specific edges from allowed edge types because it’s less efficient than not allowing an edge type.

Parameters

| edges | [in] A set of edge identifiers that must be kept intact until the destruction of the class. |

3.49.2.7 - (void) excludeNodes: (STSObjects *) nodes

Set which nodes can’t be used.
This will replace any previously specified set of excluded nodes. Should only be used to exclude the usage of specific nodes from allowed node types because it’s less efficient than not allowing a node type.

Parameters

| nodes | [in] A set of node identifiers that must be kept intact until the destruction of the class. |

3.49.2.8 - (double) getCost

Gets the cost of the shortest path.
The cost is the number of hops of the shortest path.

Returns

The cost of the shortest path.

Implements STSSinglePairShortestPath.

3.49.2.9 - (STSOidList *) getPathAsEdges

Gets the shortest path between the source node and the destination node as an ordered set of edges.

Returns

Ordered set of edge identifiers.

Implements STSSinglePairShortestPath.

3.49.2.10 - (STSOidList *) getPathAsNodes

Gets the shortest path between the source node and the destination node as an ordered set of nodes.

Returns

Ordered set of node identifiers.

Implements STSSinglePairShortestPath.

3.49.2.11 - (id) initWithSession: (STSSession *) session src: (long long) src dst: (long long) dst

Creates a new instance.
### Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>session</td>
<td>[in] Session to get the graph from and perform traversal.</td>
</tr>
<tr>
<td>src</td>
<td>[in] Source node.</td>
</tr>
<tr>
<td>dst</td>
<td>[dst] Destination node.</td>
</tr>
</tbody>
</table>

#### 3.49.2.12 - (void) setMaximumHops: (int) maxhops

Sets the maximum hops restriction.
All paths longer than the maximum hops restriction will be ignored.

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>maxhops</td>
<td>[in] The maximum hops restriction. It must be positive or zero. Zero, the default value, means unlimited.</td>
</tr>
</tbody>
</table>

The documentation for this class was generated from the following file:

- Sparksee.h

## 3.50 STSSinglePairShortestPathDijkstra Class Reference

SinglePairShortestPathDijkstra class.

Inheritance diagram for STSSinglePairShortestPathDijkstra:
Collaboration diagram for STSSinglePairShortestPathDijkstra:

Instance Methods

- (void) - run
  Executes the algorithm.
- (STSOidList *) - getPathAsNodes
  Gets the shortest path between the source node and the destination node as an ordered set of nodes.
- (STSOidList *) - getPathAsEdges
  Gets the shortest path between the source node and the destination node as an ordered set of edges.
- (double) - getCost
  Gets the cost of the shortest path.
- (id) - initWithSession:src:dst:
  Creates a new instance.
- (void) - addWeightedEdgeType:dir:attr:
  Allows for traversing edges of the given type using the given attribute as the weight.
- (void) - setUnweightedEdgeCost:
  Sets the weight assigned to the unweighted edges.
- (BOOL) - exists
  Returns TRUE if a path exists or FALSE otherwise.
- (void) - setMaximumHops:
  Sets the maximum hops restriction.
- (void) - addEdgeType:dir:
  Allows for traversing edges of the given type.
- (void) - addAllEdgeTypes:
  Allows for traversing all edge types of the graph.
- (void) - addNodeType:
Allows for traversing nodes of the given type.

- (void) - `addAllNodeTypes`
  Allows for traversing all node types of the graph.

- (void) - `excludeNodes`:
  Set which nodes can’t be used.

- (void) - `excludeEdges`:
  Set which edges can’t be used.

- (void) - `close`
  Closes the ShortestPath instance.

- (BOOL) - `isClosed`
  Check if the ShortestPath instance is closed.

### 3.50.1 Detailed Description

**SinglePairShortestPathDijkstra class.**

It solves the single-pair shortest path problem using a Dijkstra-based implementation.

It is a weighted algorithm, so it takes into account the cost of the edges to compute a minimum-cost shortest path. That is, the user may set for each edge type which attribute should be used to retrieve the cost of the edge. If no attribute is given for an edge type, this will assume the edge has a fixed cost (the default is 1). Only numerical attributes can be set as weight attributes (that is Long, Integer or Double attributes are allowed).

Check out the 'Algorithms' section in the SPARKSEE User Manual for more details on this.

**Author**

Sparsity Technologies [http://www.sparsity-technologies.com](http://www.sparsity-technologies.com)

### 3.50.2 Method Documentation

#### 3.50.2.1 - (void) `addAllEdgeTypes`: (enum STSEdgesDirection) `dir`

Allows for traversing all edge types of the graph.

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>dir</code></td>
<td>[in] Edge direction.</td>
</tr>
</tbody>
</table>

#### 3.50.2.2 - (void) `addEdgeType`: (int) `type`: (enum STSEdgesDirection) `dir`

Allows for traversing edges of the given type.

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>type</code></td>
<td>[in] Edge type.</td>
</tr>
<tr>
<td><code>dir</code></td>
<td>[in] Edge direction.</td>
</tr>
</tbody>
</table>

#### 3.50.2.3 - (void) `addNodeType`: (int) `type`

Allows for traversing nodes of the given type.

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>type</code></td>
<td>null</td>
</tr>
</tbody>
</table>
3.50.2.4 - (void) addWeightedEdgeType: (int) type dir:(enum STSEdgesDirection) dir attr:(int) attr

Allows for traversing edges of the given type using the given attribute as the weight.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>[in] Edge type.</td>
</tr>
<tr>
<td>dir</td>
<td>[in] Edge direction.</td>
</tr>
<tr>
<td>attr</td>
<td>[in] Attribute to be used as the weight. It must be a global attribute or an attribute of the given edge type.</td>
</tr>
</tbody>
</table>

3.50.2.5 - (void) close

Closes the ShortestPath instance.

It must be called to ensure the integrity of all data.

3.50.2.6 - (void) excludeEdges: (STSObjects *) edges

Set which edges can’t be used.

This will replace any previously specified set of excluded edges. Should only be used to exclude the usage of specific edges from allowed edge types because it’s less efficient than not allowing an edge type.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>edges</td>
<td>[in] A set of edge identifiers that must be kept intact until the destruction of the class.</td>
</tr>
</tbody>
</table>

3.50.2.7 - (void) excludeNodes: (STSObjects *) nodes

Set which nodes can’t be used.

This will replace any previously specified set of excluded nodes. Should only be used to exclude the usage of specific nodes from allowed node types because it’s less efficient than not allowing a node type.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nodes</td>
<td>[in] A set of node identifiers that must be kept intact until the destruction of the class.</td>
</tr>
</tbody>
</table>

3.50.2.8 - (double) getCost

Gets the cost of the shortest path.

The cost is the sum of the weights of the edges in the shortest path.

Returns

The cost of the shortest path.

Implements STSSinglePairShortestPath.

3.50.2.9 - (STSOidList *) getPathAsEdges

Gets the shortest path between the source node and the destination node as an ordered set of edges.

Returns

Ordered set of edge identifiers.

Implements STSSinglePairShortestPath.
3.50.2.10 - (STSOidList*) getPathAsNodes

Gets the shortest path between the source node and the destination node as an ordered set of nodes.

Returns

Ordered set of node identifiers.

Implements STSSinglePairShortestPath.

3.50.2.11 - (id) initWithSession: (STSSession*) session src:(long long) src dst:(long long) dst

Creates a new instance.

Parameters

| session   | [in] Session to get the graph from and perform traversal. |
| src       | [in] Source node.                                       |
| dst       | [dst] Destination node.                                |

3.50.2.12 - (void) setMaximumHops: (int) maxhops

Sets the maximum hops restriction.

All paths longer than the maximum hops restriction will be ignored.

Parameters

| maxhops | [in] The maximum hops restriction. It must be positive or zero. Zero, the default value, means unlimited. |

3.50.2.13 - (void) setUnweightedEdgeCost: (double) weight

Sets the weight assigned to the unweighted edges.

All the edges from the types added without an explicit weight attribute will get this weight. The default weight for this edges is 1.

Parameters

| weight   | [in] The weight value for unweighted edges. |

The documentation for this class was generated from the following file:

- Sparksee.h
Inheritance diagram for STSSparksee:

```
 NSObject
    ▼
STSSparksee
```

Collaboration diagram for STSSparksee:

```
 NSObject
    ▼
STSSparksee
```

Instance Methods

- (id) - initWithConfig:
  Creates a new instance.
- (STSDatabase *) - create:alias:
  Creates a new Database instance.
- (STSDatabase *) - open:readOnly:
  Opens an existing Database instance.
- (STSDatabase *) - restore:backupFile:
  Restores a Database from a backup file.
- (void) - close
  Closes the Sparksee instance.
- (BOOL) - isClosed
  Check if the Sparksee instance is closed.

Class Methods

- (NSString *) + getVersion
  Sparksee version.
3.51 STSSparksee Class Reference

3.51.1 Detailed Description

Sparksee class.
All Sparksee programs must have one single Sparksee instance to manage one or more Database instances.
This class allows for the creation of new Databases or open an existing one.

Author

Sparsity Technologies http://www.sparsity-technologies.com

3.51.2 Method Documentation

3.51.2.1 -(STSDatabase *) create: (NSString *) path alias:(NSString *) alias

Creates a new Database instance.

Parameters

- **path** [in] Database storage file.
- **alias** [in] Database alias name.

Returns

A Database instance.

Exceptions

- System.IO.IOException
  If the given file cannot be created.
- System.ApplicationException
  null

3.51.2.2 -(id) initWithConfig: (STSSparkseeConfig *) config

Creates a new instance.

Parameters

- **config** [in] Sparksee configuration.

3.51.2.3 -(STSDatabase *) open: (NSString *) path readOnly:(BOOL) readOnly

Opens an existing Database instance.

Parameters

- **path** [in] Database storage file.
- **readOnly** [in] If TRUE, open Database in read-only mode.

Returns

A Database instance.

Exceptions

- System.IO.IOException
  If the given file does not exist.
- System.ApplicationException
  null
3.51.2.4  - (STSDatabase *) restore: (NSString *) path backupFile:(NSString *) backupFile

Restores a Database from a backup file.
See the Graph class Backup method.

Parameters

<table>
<thead>
<tr>
<th>parameter</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>backupFile</td>
<td>[in] The Backup file to be restored.</td>
</tr>
</tbody>
</table>

Returns

A Database instance.

Exceptions

- **System.IO.IOException**
  - If the given file cannot be created, or the exported data file does not exist.
- **System.ApplicationException**
  - null

The documentation for this class was generated from the following file:

- Sparksee.h

3.52  STSSparkseeConfig Class Reference

Sparksee configuration class.

Inheritance diagram for STSSparkseeConfig:

```
NSObject

STSSparkseeConfig
```
3.52  STSSparkseeConfig Class Reference

Collaboration diagram for STSSparkseeConfig:

```plaintext
STSSparkseeConfig
  NSObject
```

Instance Methods

- (id) - init
  Creates a new instance.

- (int) - getExtentSize
  Gets the size of a extent.

- (void) - setExtentSize:
  Sets the size of the extents in KB.

- (int) - getExtentPages
  Gets the number of pages per extent.

- (void) - setExtentPages:
  Sets the number of pages per extent.

- (int) - getPoolFrameSize
  Gets the size of a pool frame in number of extents.

- (void) - setPoolFrameSize:
  Sets the size of a pool frame in number of extents.

- (int) - getPoolPersistentMinSize
  Gets the minimum size for the persistent pool in number of frames.

- (void) - setPoolPersistentMinSize:
  Sets the minimum size for the persistent pool in number of frames.

- (int) - getPoolPersistentMaxSize
  Gets the maximum size for the persistent pool in number of frames.

- (void) - setPoolPersistentMaxSize:
  Sets the maximum size for the persistent pool in number of frames.

- (int) - getPoolTemporaryMinSize
  Gets the minimum size for the temporary pool in number of frames.

- (void) - setPoolTemporaryMinSize:
  Sets the minimum size for the temporary pool in number of frames.

- (int) - getPoolTemporaryMaxSize
  Gets the maximum size for the temporary pool in number of frames.

- (void) - setPoolTemporaryMaxSize:
  Sets the maximum size for the temporary pool in number of frames.

- (int) - getPoolClusterSize
  Gets the number of pools in each PoolCluster.

- (void) - setPoolClusterSize:

Generated on Tue Jun 2 2015 15:49:36 for Sparksee by Doxygen
Sets the number of pools in each PoolCluster.

- (int) **getCacheMaxSize**
  Gets the maximum size for the cache (all pools) in MB.

- (void) **setCacheMaxSize**:
  Sets the maximum size for the cache (all pools) in MB.

- (NSString *) **getLicense**
  Gets the license code.

- (void) **setLicense**:
  Sets the license code.

- (NSString *) **getLogFile**
  Gets the log file.

- (void) **setLogFile**:
  Sets the log file.

- (enum STSLogLevel) **getLogLevel**
  Gets the log level.

- (void) **setLogLevel**:
  Sets the log level.

- (BOOL) **getCacheStatisticsEnabled**
  Gets whether cache statistics are enabled or disabled.

- (void) **setCacheStatisticsEnabled**:
  Enables or disables cache statistics.

- (NSString *) **getCacheStatisticsLogFile**
  Gets the cache statistics log file.

- (void) **setCacheStatisticsLogFile**:
  Sets the cache statistics log file.

- (long long) **getCacheStatisticsSnapshotTime**
  Gets the cache statistics snapshot time in microseconds.

- (void) **setCacheStatisticsSnapshotTime**:
  Sets the cache statistics snapshot time.

- (BOOL) **getRollbackEnabled**
  Gets whether the rollback is enabled or disabled.

- (void) **setRollbackEnabled**:
  Enables or disables the rollback.

- (BOOL) **getRecoveryEnabled**
  Gets whether the recovery is enabled or disabled.

- (void) **setRecoveryEnabled**:
  Enables or disables the recovery.

- (NSString *) **getRecoveryLogFile**
  Gets the recovery log file.

- (void) **setRecoveryLogFile**:
  Sets the recovery log file.

- (int) **getRecoveryCacheMaxSize**
  Gets the maximum size for the recovery log cache in extents.

- (void) **setRecoveryCacheMaxSize**:
  Sets the maximum size for the recovery log cache in extents.

- (long long) **getRecoveryCheckpointTime**
  Gets the delay time (in microseconds) between automatic checkpoints.

- (void) **setRecoveryCheckpointTime**:
  Sets the delay time (in microseconds) between automatic checkpoints.

- (BOOL) **getHighAvailabilityEnabled**
  Gets whether high availability mode is enabled or disabled.
3.52 STSSparkseeConfig Class Reference

- (void) - setHighAvailabilityEnabled:
  Enables or disables high availability mode.
- (NSString *) - getHighAvailabilityIP
  Gets the IP address and port of the instance.
- (void) - setHighAvailabilityIP:
  Sets the IP address and port of the instance.
- (NSString *) - getHighAvailabilityCoordinators
  Gets the coordinators address and port list.
- (void) - setHighAvailabilityCoordinators:
  Sets the coordinators address and port list.
- (long long) - getHighAvailabilitySynchronization
  Gets the synchronization polling time.
- (void) - setHighAvailabilitySynchronization:
  Sets the synchronization polling time.
- (long long) - getHighAvailabilityMasterHistory
  Gets the master's history log.
- (void) - setHighAvailabilityMasterHistory:
  Sets the master's history log.

3.52.1 Detailed Description

Sparksee configuration class.

If not specified, 0 means unlimited which is the maximum available. For the pools that's the total cache size. For the cache unlimited means nearly all the physical memory of the computer.

For each field, there is a default value. This value can be overrided with values from a properties file (see SparkseeProperties class). Also, this settings can be overrided calling a specific setter.

For each field, it is shown its default value and the property to override this value:

- Extent size: 4KB (‘sparksee.storage.extentsize’ at SparkseeProperties).
- Pages per extent: 1 page (‘sparksee.storage.extentpages’ at SparkseeProperties).
- Pool frame size: 1 extent (‘sparksee.io.pool.frame.size’ at SparkseeProperties).
- Minimum size for the persistent pool: 64 frames (‘sparksee.io.pool.persistent.minsize’ at SparkseeProperties).
- Maximum size for the persistent pool: 0 frames (‘sparksee.io.pool.persistent.maxsize’ at SparkseeProperties).
- Maximum size for the temporary pool: 0 frames (‘sparksee.io.pool.temporal.maxsize’ at SparkseeProperties).
- Number of pools in the pool cluster: 0 pools (‘sparksee.io.pool.clustersize’ at SparkseeProperties). 0 or 1 means the clustering is disabled.
- Maximum size for the cache (all pools): 0 MB (‘sparksee.io.cache.maxsize’ at SparkseeProperties).
- License code: "" (‘sparksee.license’ at SparkseeProperties). No license code means evaluation license.
- Log level: Info (‘sparksee.log.level’ at SparkseeProperties).
- Recovery enabled: false (‘sparksee.io.recovery’ at SparkseeProperties).
Recovery cache max size: 1MB (`sparksee.io.recovery.cachesize` at SparkseeProperties).
Recovery checkpoint time: 60 seconds [TimeUnit] (`sparksee.io.recovery.checkpointTime` at SparkseeProperties).
High-availability: false (disabled) (`sparksee.ha` at SparkseeProperties).
High-availability coordinators: "" (`sparksee.ha.coordinators` at SparkseeProperties).
High-availability IP: "" (`sparksee.ha.ip` at SparkseeProperties).
High-availability sync polling: 0 (disabled) [TimeUnit] (`sparksee.ha.sync` at SparkseeProperties).
High-availability master history: 1D (1 day) [TimeUnit] (`sparksee.ha.master.history` at SparkseeProperties).

Use of TimeUnit:

Those variables using TimeUnit allow for:

\[
<X>[D|H|M|S|s|m|u]
\]

where \(<X>\) is a number followed by an optional character which represents the unit: D for days, H for hours, M for minutes, S or s for seconds, m for milliseconds and u for microseconds. If no unit character is given, seconds are assumed.

Author

Sparsity Technologies [http://www.sparsity-technologies.com](http://www.sparsity-technologies.com)

### 3.52.2 Method Documentation

#### 3.52.2.1 - (int) getCacheMaxSize

Gets the maximum size for the cache (all pools) in MB.

**Returns**

The maximum size for the cache (all pools) in MB.

#### 3.52.2.2 - (BOOL) getCacheStatisticsEnabled

Gets whether cache statistics are enabled or disabled.

**Returns**

TRUE if cache statistics are enabled, FALSE otherwise.

#### 3.52.2.3 - (NSString *) getCacheStatisticsFile

Gets the cache statistics log file. Useless if cache statistics are disabled.

**Returns**

The cache statistics log file.

#### 3.52.2.4 - (long long) getCacheStatisticsSnapshotTime

Gets the cache statistics snapshot time in microseconds. Useless if cache statistics are disabled.

**Returns**

The cache statistics snapshot time in microseconds.
3.52.2.5 - (int) getExtentPages

Gets the number of pages per extent.

Returns
The number of pages per extent.

3.52.2.6 - (int) getExtentSize

Gets the size of a extent.

Returns
The size of a extent in KB.

3.52.2.7 - (NSString *) getHighAvailabilityCoordinators

Gets the coordinators address and port list.

Returns
The coordinators address and port list.

3.52.2.8 - (BOOL) getHighAvailabilityEnabled

Gets whether high availability mode is enabled or disabled.

Returns
TRUE if high availability mode is enabled, FALSE otherwise.

3.52.2.9 - (NSString *) getHighAvailabilityIP

Gets the IP address and port of the instance.

Returns
The IP address and port of the instance.

3.52.2.10 - (long long) getHighAvailabilityMasterHistory

Gets the master's history log.

Returns
The master's history log.

3.52.2.11 - (long long) getHighAvailabilitySynchronization

Gets the synchronization polling time.

Returns
The Synchronization polling time.
3.52.2.12 - (NSString *) getLicense

Gets the license code.

Returns

The license code.

3.52.2.13 - (NSString *) getLogFile

Gets the log file.

Returns

The log file.

3.52.2.14 - (enum STSLogLevel) getLogLevel

Gets the log level.

Returns

The LogLevel.

3.52.2.15 - (int) getPoolClusterSize

Gets the number of pools in each PoolCluster.

Returns

The number of pools in each PoolCluster.

3.52.2.16 - (int) getPoolFrameSize

Gets the size of a pool frame in number of extents.

Returns

The size of a pool frame in number of extents.

3.52.2.17 - (int) getPoolPersistentMaxSize

Gets the maximum size for the persistent pool in number of frames.

Returns

The maximum size for the persistent pool in number of frames.

3.52.2.18 - (int) getPoolPersistentMinSize

Gets the minimum size for the persistent pool in number of frames.

Returns

The minimum size for the persistent pool in number of frames.
3.52.2.19 - (int) getPoolTemporaryMaxSize

Gets the maximum size for the temporary pool in number of frames.

Returns

The maximum size for the temporary pool in number of frames.

3.52.2.20 - (int) getPoolTemporaryMinSize

Gets the minimum size for the temporary pool in number of frames.

Returns

The minimum size for the temporary pool in number of frames.

3.52.2.21 - (int) getRecoveryCacheMaxSize

Gets the maximum size for the recovery log cache in extents.

Returns

The maximum size for the recovery log cache in extents.

3.52.2.22 - (long long) getRecoveryCheckpointTime

Gets the delay time (in microseconds) between automatic checkpoints.

Returns

The delay time (in microseconds) between automatic checkpoints.

3.52.2.23 - (BOOL) getRecoveryEnabled

Gets whether the recovery is enabled or disabled.

Returns

TRUE if the recovery is enabled, FALSE otherwise.

3.52.2.24 - (NSString *) getRecoveryLogFile

Gets the recovery log file.

Returns

The recovery log file.

3.52.2.25 - (BOOL) getRollbackEnabled

Gets whether the rollback is enabled or disabled.

Returns

TRUE if the rollback is enabled, FALSE otherwise.

3.52.2.26 - (id) init

Creates a new instance.

Values are set with default values.
3.52.2.27 - (void) setCacheMaxSize: (int) coeff

Sets the maximum size for the cache (all pools) in MB.

Parameters

| coeff | [in] The maximum size for the cache (all pools) in MB. It must be non-negative. |

3.52.2.28 - (void) setCacheStatisticsEnabled: (BOOL) status

Enables or disables cache statistics.

Parameters

| status | [in] If TRUE this enables cache statistics, if FALSE this disables cache statistics. |

3.52.2.29 - (void) setCacheStatisticsFile: (NSString *) filePath

Sets the cache statistics log file.

Useless if cache statistics are disabled.

Parameters


3.52.2.30 - (void) setCacheStatisticsSnapshotTime: (long long) microSeconds

Sets the cache statistics snapshot time.

Useless if cache statistics are disabled.

Parameters


3.52.2.31 - (void) setExtentPages: (int) pages

Sets the number of pages per extent.

Parameters

| pages | [in] The number of pages. It must be at least 1 page and the page size must be greater than or equal to 4KB. |

3.52.2.32 - (void) setExtentSize: (int) kBytes

Sets the size of the extents in KB.

Parameters

| kBytes | [in] The size of an extent in KB. An extent can have a size between 4KB and 64KB, and it must be a power of 2. |

3.52.2.33 - (void) setHighAvailabilityCoordinators: (NSString *) ip

Sets the coordinators address and port list.
3.52 STSSparkseeConfig Class Reference

Parameters

| ip | [in] The coordinators address and port list. |

3.52.2.34 - (void) setHighAvailabilityEnabled: (BOOL) status

Enables or disables high availability mode.

Parameters

| status | [in] If TRUE this enables high availability mode, if FALSE this disables high availability mode. |

3.52.2.35 - (void) setHighAvailabilityIP: (NSString *) ip

Sets the IP address and port of the instance.

Parameters

| ip | [in] The IP address and port of the instance. |

3.52.2.36 - (void) setHighAvailabilityMasterHistory: (long long) filePath

Sets the master’s history log.

Parameters

| filePath | [in] The master’s history log. |

3.52.2.37 - (void) setHighAvailabilitySynchronization: (long long) microSeconds

Sets the synchronization polling time.

Parameters

| microSeconds | [in] The synchronization polling time. |

3.52.2.38 - (void) setLicense: (NSString *) key

Sets the license code.

Parameters

| key | [in] The license code. |

3.52.2.39 - (void) setLogFile: (NSString *) filePath

Sets the log file.

Parameters


3.52.2.40 - (void) setLogLevel: (enum STSLogLevel) level

Sets the log level.
### Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
</table>

#### 3.52.2.41 - (void) setPoolClusterSize: (int) pools

Sets the number of pools in each PoolCluster.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pools</td>
<td>[in]</td>
<td>The number of pools in each PoolCluster. It must be non-negative.</td>
</tr>
</tbody>
</table>

#### 3.52.2.42 - (void) setPoolFrameSize: (int) extents

Sets the size of a pool frame in number of extents.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>extents</td>
<td>[in]</td>
<td>The size of a pool frame in number of extents. It must be non-negative.</td>
</tr>
</tbody>
</table>

#### 3.52.2.43 - (void) setPoolPersistentMaxSize: (int) frames

Sets the maximum size for the persistent pool in number of frames.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>frames</td>
<td>[in]</td>
<td>The maximum size for the persistent pool in number of frames. It must be non-negative.</td>
</tr>
</tbody>
</table>

#### 3.52.2.44 - (void) setPoolPersistentMinSize: (int) frames

Sets the minimum size for the persistent pool in number of frames.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>frames</td>
<td>[in]</td>
<td>The minimum size for the persistent pool in number of frames. It must be non-negative.</td>
</tr>
</tbody>
</table>

#### 3.52.2.45 - (void) setPoolTemporaryMaxSize: (int) frames

Sets the maximum size for the temporary pool in number of frames.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>frames</td>
<td>[in]</td>
<td>The maximum size for the temporary pool in number of frames. It must be non-negative.</td>
</tr>
</tbody>
</table>

#### 3.52.2.46 - (void) setPoolTemporaryMinSize: (int) frames

Sets the minimum size for the temporary pool in number of frames.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>frames</td>
<td>[in]</td>
<td>The minimum size for the temporary pool in number of frames. It must be non-negative.</td>
</tr>
</tbody>
</table>

#### 3.52.2.47 - (void) setRecoveryCacheMaxSize: (int) extents

Sets the maximum size for the recovery log cache in extents.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>extents</td>
<td>[in]</td>
<td>The maximum size for the recovery log cache in extents.</td>
</tr>
</tbody>
</table>
Parameters

| extents   | [in] The maximum size for the recovery log cache in extents. A 0 sets the default value (extents up to 1MB). |

3.52.2.48 - (void) setRecoveryCheckpointTime: (long long) microSeconds

Sets the delay time (in microseconds) between automatic checkpoints.

Parameters

| microSeconds | [in] The delay time (in microseconds) between automatic checkpoints. A 0 forces a checkpoint after each committed transaction. |

3.52.2.49 - (void) setRecoveryEnabled: (BOOL) status

Enables or disables the recovery.

Parameters

| status | [in] If TRUE this enables the recovery, if FALSE then disables it. |

3.52.2.50 - (void) setRecoveryLogFile: (NSString *) filePath

Sets the recovery log file.

Parameters

| filePath | [in] The recovery log file. Left it empty for the default log file (same as <database_file_name>.log) |

3.52.2.51 - (void) setRollbackEnabled: (BOOL) status

Enables or disables the rollback.

Parameters

| status | [in] If TRUE this enables the rollback, if FALSE then disables it. |

The documentation for this class was generated from the following file:

- Sparksee.h

3.53 STSSparkseeProperties Class Reference

Sparksee properties file.
Class Methods

- (void) + `load`:
  Loads properties from the given file path.
- (NSString *) + `get: def`:
  Gets a property.
- (int) + `getInteger: def`:
  Gets a property as an integer.
- (BOOL) + `getBoolean: def`:
  Gets a property as a boolean.
- (long long) + `getTimeUnit: def`:
  Gets a property as a time unit.

3.53.1 Detailed Description

Sparksee properties file.

This class is implemented as a singleton, so all public methods are static.

It allows for getting the property values stored in a properties file. A properties file is a file where there is one line per property. A property is defined by a key and a value as follows: key=value
By default, this loads properties from the file `./sparksee.cfg`. The user may choose to load a different file by calling the method `Load()`.

If the default properties file or the one loaded by the user do not exist, then this behaves as loading an empty properties file.

### 3.53.2 Method Documentation

#### 3.53.2.1 + (NSString *) get: (NSString *) key def:(NSString *) def

Gets a property.

**Parameters**

- `key` [in] The name of the property to lookup.
- `def` [in] Default value to be returned in case there is no property with the name `key`.

**Returns**

The value of the property, or `def` if the key is not found.

#### 3.53.2.2 + (BOOL) getBoolean: (NSString *) key def:(BOOL) def

Gets a property as a boolean.

**Parameters**

- `key` [in] The name of the property to lookup.
- `def` [in] Default value to be returned in case there is no property with the name `key`.

**Returns**

The property value, or `def` if the key is not found or in case of error.

#### 3.53.2.3 + (int) getInteger: (NSString *) key def:(int) def

Gets a property as an integer.

**Parameters**

- `key` [in] The name of the property to lookup.
- `def` [in] Default value to be returned in case there is no property with the name `key`.

**Returns**

The property value, or `def` if the key is not found or in case of error.

#### 3.53.2.4 + (long long) getTimeUnit: (NSString *) key def:(long long) def

Gets a property as a time unit.

A time unit is a string representation of a time duration with a time unit such as '10s' or '3H'.

Valid format for the string representation: Blanks at the begining or at the end are ignored. No blanks are allowed between the time duration and the unit time.

Allowed time units: 'D' for days, 'H' for hours, 'M' for minutes, 'S' or 's' for seconds, 'm' for milliseconds and 'u' for microseconds.

There is a special case: If no time unit is given, seconds is the default. So, '10' means 10 seconds.
3.54 STSStringList Class Reference

Parameters

<table>
<thead>
<tr>
<th>key</th>
<th>[in] The name of the property to lookup.</th>
</tr>
</thead>
<tbody>
<tr>
<td>def</td>
<td>[in] The default value (in microseconds) to be returned in case there is no property with the name key.</td>
</tr>
</tbody>
</table>

Returns

The time duration in microseconds, or def if the key is not found or in case of error.

3.53.2.5 + (void) load: (NSString *) path

Loads properties from the given file path.

Parameters

| path | [in] File path to load properties from. |

The documentation for this class was generated from the following file:

- Sparksee.h

3.54 STSStringList Class Reference

String list.

Inheritance diagram for STSStringList:

```
NSObject

STSStringList
```
Collaboration diagram for STSStringList:

```
+-----------------+        +-----------------+
|                |        |                |
|      STSSStringList          |        | NSObject       |
|                |        |                |
```

Instance Methods

- `(int)` - `count`
  Number of elements in the list.
- `(id)` - `init`
  Constructor.
- `(void)` - `add:`
  Adds a String at the end of the list.
- `(void)` - `clear`
  Clears the list.
- `(id)` - `initWithArray:`
  Creates a new StringList instance from the given array.
- `(id)` - `initWithNSEnumerator:`
  Creates a new StringList instance from the given NSEnumerator.
- `(STSStringListIterator *)` - `iterator`
  Gets a new StringListIterator.

3.54.1 Detailed Description

String list.

It stores a String (unicode) list.

Use StringListIterator to access all elements into this collection.

Author

Sparsity Technologies http://www.sparsity-technologies.com

3.54.2 Method Documentation

3.54.2.1 `(void)` - add: `(NSString *)` `str`

Adds a String at the end of the list.

Parameters

3.54.2.2 - (int) count

Number of elements in the list.

Returns

Number of elements in the list.

3.54.2.3 - (id) init

Constructor.
This creates an empty list.
The documentation for this class was generated from the following file:

- Sparksee.h

3.55 STSStringListIterator Class Reference

StringList iterator class.

Inheritance diagram for STSStringListIterator:

```
    NSObject
      ↓
    STSStringListIterator
```
Collaboration diagram for STSStringListIterator:

```
<table>
<thead>
<tr>
<th>NSObject</th>
</tr>
</thead>
<tbody>
<tr>
<td>STSStringList</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>theParent</td>
</tr>
<tr>
<td>STSStringListIterator</td>
</tr>
</tbody>
</table>
```

Instance Methods

- `(NSString *) next`
  
  Moves to the next element.

- `(BOOL) hasNext`
  
  Gets if there are more elements.

3.55.1 Detailed Description

StringList iterator class.

Iterator to traverse all the strings into a StringList instance.

Author

Sparsity Technologies [http://www.sparsity-technologies.com](http://www.sparsity-technologies.com)

3.55.2 Method Documentation

3.55.2.1 - (BOOL) hasNext

Gets if there are more elements.

Returns

TRUE if there are more elements, FALSE otherwise.

3.55.2.2 - (NSString *) next

Moves to the next element.
Returns

The next element.

The documentation for this class was generated from the following file:

• Sparksee.h

3.56 STSStrongConnectivity Class Reference

StrongConnectivity class.

Inheritance diagram for STSStrongConnectivity:
Collaboration diagram for STSStrongConnectivity:

```
NSObject

<table>
<thead>
<tr>
<th>STSConnectivity</th>
</tr>
</thead>
<tbody>
<tr>
<td>STSStrongConnectivity</td>
</tr>
</tbody>
</table>
```

### Instance Methods

- (void) - addEdgeType:dir:
  
  Allows connectivity through edges of the given type.

- (void) - addAllEdgeTypes:
  
  Allows connectivity through all edge types of the graph.

- (void) - addNodeType:
  
  Allows connectivity through nodes of the given type.

- (void) - addAllNodeTypes
  
  Allows connectivity through all node types of the graph.

- (void) - excludeNodes:
  
  Set which nodes can’t be used.

- (void) - excludeEdges:
  
  Set which edges can’t be used.

- (STSConnectedComponents *) - getConnectedComponents
  
  Returns the results generated by the execution of the algorithm.

- (void) - run
  
  Runs the algorithm in order to find the connected components.

- (void) - setMaterializedAttribute:
  
  Creates a new common attribute type for all node types in the graph in order to store, persistently, the results related to the connected components found while executing this algorithm.

- (void) - close
  
  Closes the Connectivity instance.

- (BOOL) - isClosed
  
  Check if the Connectivity instance is closed.

### 3.56.1 Detailed Description

**StrongConnectivity class.**

Any class implementing this abstract class can be used to solve the problem of finding strongly connected components in a directed graph.
It consists in finding components where every pair \((u,v)\) of nodes contained in it has a path from \(u\) to \(v\) using the specified direction for each edge type.

It is possible to set some restrictions after constructing a new instance of this class and before running it in order to limit the results.

After the execution, we can retrieve the results stored in an instance of the ConnectedComponents class using the GetConnectedComponents method.

Check out the 'Algorithms' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies [http://www.sparsity-technologies.com](http://www.sparsity-technologies.com)

### 3.56.2 Method Documentation

#### 3.56.2.1 - (void) addAllEdgeTypes: (enum STSEdgesDirection) \(dir\)

Allows connectivity through all edge types of the graph.

**Parameters**

<table>
<thead>
<tr>
<th>(dir)</th>
<th>[in] Edge direction.</th>
</tr>
</thead>
</table>

#### 3.56.2.2 - (void) addEdgeType: (int) \(type\) dir:(enum STSEdgesDirection) \(dir\)

Allows connectivity through edges of the given type.

**Parameters**

<table>
<thead>
<tr>
<th>(type)</th>
<th>[in] Edge type.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>(dir)</th>
<th>[in] Edge direction.</th>
</tr>
</thead>
</table>

#### 3.56.2.3 - (void) addNodeType: (int) \(t\)

Allows connectivity through nodes of the given type.

**Parameters**

<table>
<thead>
<tr>
<th>(t)</th>
<th>null</th>
</tr>
</thead>
</table>

#### 3.56.2.4 - (void) close

Closes the Connectivity instance.

It must be called to ensure the integrity of all data.

#### 3.56.2.5 - (void) excludeEdges: (STSObjects *) \(edges\)

Set which edges can’t be used.

This will replace any previously specified set of excluded edges. Should only be used to exclude the usage of specific edges from allowed edge types because it’s less efficient than not allowing an edge type.

**Parameters**

| \(edges\) | [in] A set of edge identifiers that must be kept intact until the destruction of the class. |
### 3.56.2.6 - (void) excludeNodes: (STSObjects *) nodes

Set which nodes can’t be used.

This will replace any previously specified set of excluded nodes. Should only be used to exclude the usage of specific nodes from allowed node types because it’s less efficient than not allowing a node type.

**Parameters**

| nodes | [in] A set of node identifiers that must be kept intact until the destruction of the class. |

### 3.56.2.7 - (STSConnectedComponents *) getConnectedComponents

Returns the results generated by the execution of the algorithm.

These results contain information related to the connected components found as the number of different components, the set of nodes contained in each component or many other data.

**Returns**

Returns an instance of the class ConnectedComponents which contain information related to the connected components found.

### 3.56.2.8 - (void) run

Runs the algorithm in order to find the connected components.

This method can be called only once.

Implemented in STSStrongConnectivityGabow, and STSWeakConnectivityDFS.

### 3.56.2.9 - (void) setMaterializedAttribute: (NSString *) attributeName

Creates a new common attribute type for all node types in the graph in order to store, persistently, the results related to the connected components found while executing this algorithm.

Whenever the user wants to retrieve the results, even when the graph has been closed and opened again, it is only necessary to create a new instance of the class ConnectedComponents indicating the graph and the name of the common attribute type which stores the results. This instance will have all the information related to the connected components found in the moment of the execution of the algorithm that stored this data.

It is possible to run the algorithm without specifying this parameter in order to avoid materializing the results of the execution.

**Parameters**

| attributeName | [in] The name of the common attribute type for all node types in the graph which will store persistently the results generated by the execution of the algorithm. |

The documentation for this class was generated from the following file:

- Sparksee.h

### 3.57 STSStrongConnectivityGabow Class Reference

This class can be used to solve the problem of finding strongly connected components in a directed graph.
Inheritance diagram for STSStrongConnectivityGabow:

```
  NSObject
  |    |
  v    v
STSConnectivity
  |    |
  v    v
STSStrongConnectivity
  |    |
  v    v
STSStrongConnectivityGabow
```

Collaboration diagram for STSStrongConnectivityGabow:

```
  NSObject
  |    |
  v    v
STSConnectivity
  |    |
  v    v
STSStrongConnectivity
  |    |
  v    v
STSStrongConnectivityGabow
```
Instance Methods

- (id) - initWithSession:
  Creates a new instance of StrongConnectivityGabow.

- (void) - run
  Executes the algorithm.

- (void) - addEdgeType:dir:
  Allows connectivity through edges of the given type.

- (void) - addAllEdgeTypes:
  Allows connectivity through all edge types of the graph.

- (void) - addNodeType:
  Allows connectivity through nodes of the given type.

- (void) - addAllNodeTypes
  Allows connectivity through all node types of the graph.

- (void) - excludeNodes:
  Set which nodes can't be used.

- (void) - excludeEdges:
  Set which edges can't be used.

- (STSConnectedComponents *) - getConnectedComponents
  Returns the results generated by the execution of the algorithm.

- (void) - setMaterializedAttribute:
  Creates a new common attribute type for all node types in the graph in order to store, persistently, the results related to the connected components found while executing this algorithm.

- (void) - close
  Closes the Connectivity instance.

- (BOOL) - isClosed
  Check if the Connectivity instance is closed.

3.57.1 Detailed Description

This class can be used to solve the problem of finding strongly connected components in a directed graph.

It consists in finding components where every pair (u,v) of nodes contained in it has a path from u to v using the specified direction for each edge type. This implementation is based on the Gabow algorithm.

It is possible to set some restrictions after constructing a new instance of this class and before running it in order to limit the results.

After the execution, we can retrieve the results stored in an instance of the ConnectedComponents class using the GetConnectedComponents method.

Check out the 'Algorithms' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies [http://www.sparsity-technologies.com](http://www.sparsity-technologies.com)

3.57.2 Method Documentation

3.57.2.1 - (void) addAllEdgeTypes: (enum STSEdgesDirection) dir

Allows connectivity through all edge types of the graph.

Parameters

| dir   | [in] Edge direction. |
3.57.2.2 - (void) addEdgeType: (int) type (enum STSEdgesDirection) dir

Allows connectivity through edges of the given type.

Parameters

| type | [in] Edge type. |
| dir  | [in] Edge direction. |

3.57.2.3 - (void) addNodeType: (int) t

Allows connectivity through nodes of the given type.

Parameters

| t | null |

3.57.2.4 - (void) close

Closes the Connectivity instance. It must be called to ensure the integrity of all data.

3.57.2.5 - (void) excludeEdges: (STSObjects *) edges

Set which edges can’t be used. This will replace any previously specified set of excluded edges. Should only be used to exclude the usage of specific edges from allowed edge types because it’s less efficient than not allowing an edge type.

Parameters

| edges | [in] A set of edge identifiers that must be kept intact until the destruction of the class. |

3.57.2.6 - (void) excludeNodes: (STSObjects *) nodes

Set which nodes can’t be used. This will replace any previously specified set of excluded nodes. Should only be used to exclude the usage of specific nodes from allowed node types because it’s less efficient than not allowing a node type.

Parameters

| nodes | [in] A set of node identifiers that must be kept intact until the destruction of the class. |

3.57.2.7 - (STSConnectedComponents *) getConnectedComponents

Returns the results generated by the execution of the algorithm. These results contain information related to the connected components found as the number of different components, the set of nodes contained in each component or many other data.

Returns

Returns an instance of the class ConnectedComponents which contain information related to the connected components found.

3.57.2.8 - (id) initWithSession: (STSSession *) session

 Creates a new instance of StrongConnectivityGabow.
After creating this instance is required to indicate the set of edge types and the set of node types which will be navigated through while traversing the graph in order to find the strong connected components.

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>session</td>
<td>[in] Session to get the graph from and calculate the connectivity</td>
</tr>
</tbody>
</table>

3.57.2.9 - (void) setMaterializedAttribute: (NSString *) attributeName

Creates a new common attribute type for all node types in the graph in order to store, persistently, the results related to the connected components found while executing this algorithm.

Whenever the user wants to retrieve the results, even when the graph has been closed and opened again, it is only necessary to create a new instance of the class ConnectedComponents indicating the graph and the name of the common attribute type which stores the results. This instance will have all the information related to the connected components found in the moment of the execution of the algorithm that stored this data.

It is possible to run the algorithm without specifying this parameter in order to avoid materializing the results of the execution.

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>attributeName</td>
<td>[in] The name of the common attribute type for all node types in the graph which will store persistently the results generated by the execution of the algorithm.</td>
</tr>
</tbody>
</table>

The documentation for this class was generated from the following file:

- Sparksee.h

3.58 STSTextStream Class Reference

TextStream class.

Inheritance diagram for STSTextStream:
Collaboration diagram for STSTextStream:

```
NSObject
  STSTextStream
```

Instance Methods

- (id) - initWithAppend:
  
  Creates a new instance.

- (BOOL) - IsNull
  
  Returns TRUE if the stream is not available.

- (void) - writeString:
  
  Write data to the stream.

- (NSString *) - readString:
  
  Read data from the stream.

- (void) - close
  
  Closes the TextStream instance.

- (BOOL) - isClosed
  
  Check if the TextStream instance is closed.

3.58.1 Detailed Description

TextStream class.

It allows for reading and writting Text attribute values.

It is very important to close the stream once no more reading or writting operations will be performed to ensure data is successfully stored.

Whereas string attributes are set and got using the Value class, text attributes are operated using a stream pattern.

Use of TextStream for writing: (i) Create a TextStream instance and (ii) set the stream for a text attribute of a node or edge instance with the graph SetAttributeText method. Once the set attribute text has been done, (iii) perform as many write operations as you need to the TextStream instance. Lastly, (iv) execute Close to flush and close the stream.

Use of TextStream for reading: (i) Get the stream of a text attribute of a node or edge instance with the GetAttributeText graph method. Once you have the TextStream instance, (ii) you can execute Read operations to read from the stream. (iii) The end of the stream is reached when Read returns 0. Finally, (iv) execute Close to close stream resources.

Check out the ‘Attributes and values’ section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies [http://www.sparsity-technologies.com](http://www.sparsity-technologies.com)
3.58.2 Method Documentation

3.58.2.1 - (void) close

Closes the TextStream instance.
It must be called to ensure the integrity of all data.

3.58.2.2 - (id) initWithAppend: (BOOL) append

Creates a new instance.
A TextStream only can be created by the user to write data.

Parameters

| append | [in] If TRUE, the it is created in append mode to write from the end of the stream, otherwise it is created to write from the begining of the stream. |

3.58.2.3 - (BOOL) IsNull

Returns TRUE if the stream is not available.
It returns for reading or writing data.

Returns

FALSE if the stream is ready

3.58.2.4 - (NSString *) readString: (int) length

Read data from the stream.
Reads a certain amount of characters (<= length). If the result string is empty, there is no more data to be read from the stream.

Parameters

| length | [in] Maximum length to be read. It must be > 0. |

Returns

Returns the string readed.

3.58.2.5 - (void) writeString: (NSString *) inStr

Write data to the stream.

Parameters

| inStr | [in] The source string. |

The documentation for this class was generated from the following file:

- Sparksee.h

3.59 STSTraversal Class Reference

Traversal class.
Inheritance diagram for STSTraversal:

```
Inheritance diagram for STSTraversal:

    NSObject →
    |  |
    |  |
  STSTraversal →
  |             |
  |             |
STSTraversalBFS  STSTraversalDFS
```

Collaboration diagram for STSTraversal:

```
Collaboration diagram for STSTraversal:

    NSObject →
    |  |
    |  |
  STSTraversal
```

Instance Methods

- (void) - addEdgeType:dir:
  Allows for traversing edges of the given type.
- (void) - addAllEdgeTypes:
  Allows for traversing all edge types of the graph.
- (void) - addNodeType:
  Allows for traversing nodes of the given type.
- (void) - addAllNodeTypes
  Allows for traversing all node types of the graph.
- (void) - excludeNodes:
  Set which nodes can't be used.
- (void) - excludeEdges:
  Set which edges can't be used.
- (long long) - next
Gets the next object of the traversal.

- (BOOL) - hasNext
  Gets if there are more objects to be traversed.

- (int) - getCurrentDepth
  Returns the depth of the current node.

- (void) - setMaximumHops:
  Sets the maximum hops restriction.

- (void) - close
  Closes the Traversal instance.

- (BOOL) - isClosed
  Check if the Traversal instance is closed.

### 3.59.1 Detailed Description

Traversal class.

Any class implementing this abstract class can be used to traverse a graph.

Once the instance has been created and the allowed node and edge types has been set, it can be used as an iterator, retrieving the next object identifier of the traversal until there are no more.

Check out the 'Algorithms' section in the SPARKSEE User Manual for more details on this.

**Author**

Sparsity Technologies [http://www.sparsity-technologies.com](http://www.sparsity-technologies.com)

### 3.59.2 Method Documentation

#### 3.59.2.1 - (void) addAllEdgeTypes: (enum STSEdgesDirection) dir

Allows for traversing all edge types of the graph.

**Parameters**

<table>
<thead>
<tr>
<th>dir</th>
<th>[in] Edge direction.</th>
</tr>
</thead>
</table>

#### 3.59.2.2 - (void) addEdgeType: (int) type dir:(enum STSEdgesDirection) dir

Allows for traversing edges of the given type.

**Parameters**

<table>
<thead>
<tr>
<th>type</th>
<th>[in] Edge type.</th>
</tr>
</thead>
<tbody>
<tr>
<td>dir</td>
<td>[in] Edge direction.</td>
</tr>
</tbody>
</table>

#### 3.59.2.3 - (void) addNodeType: (int) type

Allows for traversing nodes of the given type.

**Parameters**

<table>
<thead>
<tr>
<th>type</th>
<th>null</th>
</tr>
</thead>
</table>

#### 3.59.2.4 - (void) close

Closes the Traversal instance.
It must be called to ensure the integrity of all data.

3.59.2.5 - (void) excludeEdges: (STSObjects *) edges

Set which edges can't be used.
This will replace any previously specified set of excluded edges. Should only be used to exclude the usage of specific edges from allowed edge types because it's less efficient than not allowing an edge type.

Parameters

| edges | [in] A set of edge identifiers that must be kept intact until the destruction of the class. |

3.59.2.6 - (void) excludeNodes: (STSObjects *) nodes

Set which nodes can't be used.
This will replace any previously specified set of excluded nodes. Should only be used to exclude the usage of specific nodes from allowed node types because it's less efficient than not allowing a node type.

Parameters

| nodes | [in] A set of node identifiers that must be kept intact until the destruction of the class. |

3.59.2.7 - (int) getCurrentDepth

Returns the depth of the current node.
That is, it returns the depth of the node returned in the last call to Next().

Returns

The depth of the current node.

Implemented in STSTraversalDFS, and STSTraversalBFS.

3.59.2.8 - (BOOL) hasNext

Gets if there are more objects to be traversed.

Returns

TRUE if there are more objects, FALSE otherwise.

Implemented in STSTraversalDFS, and STSTraversalBFS.

3.59.2.9 - (long long) next

Gets the next object of the traversal.

Returns

A node or edge identifier.

Implemented in STSTraversalDFS, and STSTraversalBFS.

3.59.2.10 - (void) setMaximumHops: (int) maxhops

Sets the maximum hops restriction.
All paths longer than the maximum hops restriction will be ignored.
Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>maxhops</td>
<td>[in] The maximum hops restriction. It must be positive or zero. Zero, the default value, means unlimited.</td>
</tr>
</tbody>
</table>

The documentation for this class was generated from the following file:

- Sparksee.h

3.60  STSTraversalBFS Class Reference

Breadth-First Search implementation of Traversal.

Inheritance diagram for STSTraversalBFS:
Instance Methods

- (long long) - next
  Gets the next object of the traversal.

- (BOOL) - hasNext
  Gets if there are more objects to be traversed.

- (int) - getCurrentDepth
  Returns the depth of the current node.

- (id) - initWithSession:node:
  Creates a new instance.

- (void) - addEdgeType:dir:
  Allows for traversing edges of the given type.

- (void) - addAllEdgeTypes:
  Allows for traversing all edge types of the graph.

- (void) - addNodeType:
  Allows for traversing nodes of the given type.

- (void) - addAllNodeTypes
  Allows for traversing all node types of the graph.

- (void) - excludeNodes:
  Set which nodes can't be used.

- (void) - excludeEdges:
  Set which edges can't be used.

- (void) - setMaximumHops:
  Sets the maximum hops restriction.

- (void) - close
  Closes the Traversal instance.

- (BOOL) - isClosed
  Check if the Traversal instance is closed.

3.60.1 Detailed Description

Breadth-First Search implementation of Traversal.
Starting from a source node, it visits all its neighbors at distance 1, then all its neighbors at distance 2, and so on.
Check out the 'Algorithms' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com

3.60.2 Method Documentation

3.60.2.1 - (void) addAllEdgeTypes: (enum STSEdgesDirection) dir
Allows for traversing all edge types of the graph.

Parameters

| dir | [in] Edge direction. |

3.60.2.2 - (void) addEdgeType: (int) type:(enum STSEdgesDirection) dir
Allows for traversing edges of the given type.

Check out the 'Algorithms' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com
Parameters

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>type</strong></td>
<td>[in] Edge type.</td>
</tr>
<tr>
<td><strong>dir</strong></td>
<td>[in] Edge direction.</td>
</tr>
</tbody>
</table>

3.60.2.3  - (void) addNodeType: (int) type

Allows for traversing nodes of the given type.

Parameters

| **type** | null |

3.60.2.4  - (void) close

Closes the Traversal instance.

It must be called to ensure the integrity of all data.

3.60.2.5  - (void) excludeEdges: (STS::Objects ∗) edges

Set which edges can’t be used.

This will replace any previously specified set of excluded edges. Should only be used to exclude the usage of specific edges from allowed edge types because it’s less efficient than not allowing an edge type.

Parameters

| **edges** | [in] A set of edge identifiers that must be kept intact until the destruction of the class. |

3.60.2.6  - (void) excludeNodes: (STS::Objects ∗) nodes

Set which nodes can’t be used.

This will replace any previously specified set of excluded nodes. Should only be used to exclude the usage of specific nodes from allowed node types because it’s less efficient than not allowing a node type.

Parameters

| **nodes** | [in] A set of node identifiers that must be kept intact until the destruction of the class. |

3.60.2.7  - (int) getCurrentDepth

Returns the depth of the current node.

That is, it returns the depth of the node returned in the last call to Next().

Returns

The depth of the current node.

Implements STSTraversal.

3.60.2.8  - (BOOL) hasNext

Gets if there are more objects to be traversed.

Returns

TRUE if there are more objects, FALSE otherwise.

Implements STSTraversal.
3.60.9 - (id) initWithSession: (STSSession *) session node:(long long) node

Creates a new instance.

Parameters

<table>
<thead>
<tr>
<th>session</th>
<th>[in] Session to get the graph from and perform traversal.</th>
</tr>
</thead>
<tbody>
<tr>
<td>node</td>
<td>[in] Node to start traversal from.</td>
</tr>
</tbody>
</table>

3.60.10 - (long long) next

Gets the next object of the traversal.

Returns

A node or edge identifier.

Implements STSTraversal.

3.60.11 - (void) setMaximumHops: (int) maxhops

Sets the maximum hops restriction.

All paths longer than the maximum hops restriction will be ignored.

Parameters

| maxhops | [in] The maximum hops restriction. It must be positive or zero. Zero, the default value, means unlimited. |

The documentation for this class was generated from the following file:

- Sparksee.h

3.61 STSTraversalDFS Class Reference

Depth-First Search (DFS) implementation of Traversal.

Inheritance diagram for STSTraversalDFS:
Collaboration diagram for STSTraversalDFS:

```
 Nicht

 NSObject

 STSTraversal

 STSTraversalDFS
```

Instance Methods

- (long long) - next
  
  Gets the next object of the traversal.

- (BOOL) - hasNext
  
  Gets if there are more objects to be traversed.

- (int) - getCurrentDepth
  
  Returns the depth of the current node.

- (id) - initWithSession:node:
  
  Creates a new instance.

- (void) - addEdgeType:dir:
  
  Allows for traversing edges of the given type.

- (void) - addAllEdgeTypes:
  
  Allows for traversing all edge types of the graph.

- (void) - addNodeType:
  
  Allows for traversing nodes of the given type.

- (void) - addAllNodeTypes
  
  Allows for traversing all node types of the graph.

- (void) - excludeNodes:
  
  Set which nodes can't be used.

- (void) - excludeEdges:
  
  Set which edges can't be used.

- (void) - setMaximumHops:
  
  Sets the maximum hops restriction.

- (void) - close
  
  Closes the Traversal instance.

- (BOOL) - isClosed
  
  Check if the Traversal instance is closed.
3.61 STSTraversalDFS Class Reference

3.61.1 Detailed Description

Depth-First Search (DFS) implementation of Traversal.
Starting from a source or root node, it visits as far as possible along each branch before backtracking.
Check out the 'Algorithms' section in the SPARKSEE User Manual for more details on this.

Author
Sparsity Technologies http://www.sparsity-technologies.com

3.61.2 Method Documentation

3.61.2.1 - (void) addAllEdgeTypes: (enum STSEdgesDirection) dir

Allows for traversing all edge types of the graph.

Parameters
| dir  | [in] Edge direction. |

3.61.2.2 - (void) addEdgeType: (int) type:(enum STSEdgesDirection) dir

Allows for traversing edges of the given type.

Parameters
| type | [in] Edge type. |
| dir  | [in] Edge direction. |

3.61.2.3 - (void) addNodeType: (int) type

Allows for traversing nodes of the given type.

Parameters
| type | null |

3.61.2.4 - (void) close

Closes the Traversal instance.
It must be called to ensure the integrity of all data.

3.61.2.5 - (void) excludeEdges: (STSObjects *) edges

Set which edges can’t be used.
This will replace any previously specified set of excluded edges. Should only be used to exclude the usage of specific edges from allowed edge types because it’s less efficient than not allowing an edge type.

Parameters
| edges | [in] A set of edge identifiers that must be kept intact until the destruction of the class. |

3.61.2.6 - (void) excludeNodes: (STSObjects *) nodes

Set which nodes can’t be used.
This will replace any previously specified set of excluded nodes. Should only be used to exclude the usage of
specific nodes from allowed node types because it’s less efficient than not allowing a node type.

Parameters

| nodes | [in] A set of node identifiers that must be kept intact until the destruction of the class. |

3.61.2.7 - (int) getCurrentDepth

Returns the depth of the current node.
That is, it returns the depth of the node returned in the last call to Next().

Returns

The depth of the current node.

Implements STSTraversal.

3.61.2.8 - (BOOL) hasNext

Gets if there are more objects to be traversed.

Returns

TRUE if there are more objects, FALSE otherwise.

Implements STSTraversal.

3.61.2.9 - (id) initWithSession: (STSSession *) session node:(long long) node

Creates a new instance.

Parameters

| session | [in] Session to get the graph from and perform traversal. |
| node    | [in] Node to start traversal from. |

3.61.2.10 - (long long) next

Gets the next object of the traversal.

Returns

A node or edge identifier.

Implements STSTraversal.

3.61.2.11 - (void) setMaximumHops: (int) maxhops

Sets the maximum hops restriction.
All paths longer than the maximum hops restriction will be ignored.

Parameters

| maxhops | [in] The maximum hops restriction. It must be positive or zero. Zero, the default value, means unlimited. |

The documentation for this class was generated from the following file:

- Sparksee.h
Type data class.

Inheritance diagram for STSType:

Collaboration diagram for STSType:

Instance Methods

- (int) - getId
  Gets the Sparksee type identifier.
- (enum STSObjectType) - getObjectType
  Gets the object type.
- (NSString *) - getName
  Gets the unique type name.
- (long long) - getNumObjects
  Gets the number of objects belonging to the type.
- (BOOL) - getIsDirected
  Gets if this is a directed edge type.
- (BOOL) - getIsRestricted
  Gets if this is a restricted edge type.
- (BOOL) - getAreNeighborsIndexed
  Gets if this is an edge type with neighbors index.
* (int) - **getRestrictedFrom**
  
  Gets the tail or source type identifier for restricted edge types.

* (int) - **getRestrictedTo**
  
  Gets the head or target type identifier for restricted edge types.

### Class Methods

* (int) + **getInvalidType**
  
  Invalid type identifier.

* (int) + **getGlobalType**
  
  Global type identifier.

* (int) + **getNodeType**
  
  Identifier for all nodeType attributes.

* (int) + **getEdgesType**
  
  Identifier for all edgeType attributes.

### 3.62.1 Detailed Description

Type data class.

It contains information about a node or edge type.

**Author**

Sparsity Technologies [http://www.sparsity-technologies.com](http://www.sparsity-technologies.com)

### 3.62.2 Method Documentation

#### 3.62.2.1 - (BOOL) getAreNeighborsIndexed

Gets if this is an edge type with neighbors index.

**Returns**

TRUE for edges types with neighbors index, FALSE otherwise.

#### 3.62.2.2 - (int) getId

Gets the Sparksee type identifier.

**Returns**

The Sparksee type identifier.

#### 3.62.2.3 - (BOOL) getIsDirected

Gets if this is a directed edge type.

**Returns**

TRUE for directed edge types, FALSE otherwise.
3.62.4 - (BOOL) getIsRestricted

Gets if this is a restricted edge type.

Returns

TRUE for restricted edge types, FALSE otherwise.

3.62.5 - (NSString *) getName

Gets the unique type name.

Returns

The unique type name.

3.62.6 - (long long) getNumObjects

Gets the number of objects belonging to the type.

Returns

The number of objects belonging to the type.

3.62.7 - (enum STSObjectType) getObjectType

Gets the object type.

Returns

The object type.

3.62.8 - (int) getRestrictedFrom

Gets the tail or source type identifier for restricted edge types.

Returns

For restricted edge types, the tail or source type identifier, the Type InvalidType otherwise.

3.62.9 - (int) getRestrictedTo

Gets the head or target type identifier for restricted edge types.

Returns

For restricted edge types, the head or target type identifier, the Type InvalidType otherwise.

The documentation for this class was generated from the following file:

• Sparksee.h
Inheritance diagram for STSTypeExporter:

Collaboration diagram for STSTypeExporter:

Instance Methods

- (void) - registerListener:
  Registers a new listener.
- (void) - run
  Runs export process.
- (void) - setRowWriter:
  Sets the output data destination.
- (void) - setGraph:
  Sets the graph that will be exported.
- (void) - setType:
  Sets the type to be exported.
- (void) - setAttributes:
  Sets the list of Attributes.
- (void) - setFrequency:
3.63 STSTypeExporter Class Reference

Sets the frequency of listener notification.

- (void) - setHeader:
  Sets the presence of a header row.

3.63.1 Detailed Description

Base TypeExporter class.

Base class to export a node or edge type from a graph using a RowWriter.

TypeExporterListener can be registered to receive information about the progress of the export process by means of TypeExporterEvent. The default frequency of notification to listeners is 100000.

By default no header row is created.

Check out the 'Data export' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com

3.63.2 Method Documentation

3.63.2.1 - (void) registerListener: (STSTypeExporterListener *) tel

Registers a new listener.

Parameters

| tel | [in] TypeExporterListener to be registered. |

3.63.2.2 - (void) run

Runs export process.

Exceptions

- System.IO.IOException If bad things happen writing to the RowWriter.
- System.ApplicationException null

Implemented in STSEdgeTypeExporter, and STSNodeTypeExporter.

3.63.2.3 - (void) setAttributes: (STSAttributeList *) attrs

Sets the list of Attributes.

Parameters

| attrs | [in] Attribute identifiers to be exported |

3.63.2.4 - (void) setFrequency: (int) freq

Sets the frequency of listener notification.

Parameters

| freq | [in] Frequency in number of rows managed to notify progress to all listeners |
3.63.2.5  - (void) setGraph: (STSGraph *) graph
Sets the graph that will be exported.

Parameters

| graph | [in] Graph. |

3.63.2.6  - (void) setHeader: (BOOL) header
Sets the presence of a header row.

Parameters

| header | [in] If TRUE, a header row is dumped with the name of the attributes. |

3.63.2.7  - (void) setRowWriter: (STSRowWriter *) rw
Sets the output data destination.

Parameters

| rw | [in] Input RowWriter. |

3.63.2.8  - (void) setType: (int) type
Sets the type to be exported.

Parameters

| type | [in] Type identifier. |

The documentation for this class was generated from the following file:

- Sparksee.h

3.64  STSTypeExporterEvent Class Reference

Provides information about the progress of an TypeExproter instance.

Inheritance diagram for STSTypeExporterEvent:

```
  NSObject
      ▼
       ↘
        STSTypeExporterEvent
```
Collaboration diagram for STSTypeExporterEvent:

Instance Methods

• (int) - `getTypeId`
  
  Gets the type identifier.

• (long long) - `getCount`
  
  Gets the current number of objects exported.

• (long long) - `getTotal`
  
  Gets the total number of objects exported.

• (BOOL) - `isLast`
  
  Gets if this is the last event or not.

3.64.1 Detailed Description

Provides information about the progress of an TypeExproter instance.

Check out the 'Data export' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com

3.64.2 Method Documentation

3.64.2.1 - (long long) getCount

Gets the current number of objects exported.

Returns

The current number of objects exported.

3.64.2.2 - (long long) getTotal

Gets the total number of objects exported.

Returns

The total number of objects exported.
3.64.2.3 - (int) getTypeId

Gets the type identifier.

Returns

The type identifier.

3.64.2.4 - (BOOL) isLast

Gets if this is the last event or not.

Returns

TRUE if this is the last event, FALSE otherwise.

The documentation for this class was generated from the following file:

- Sparksee.h

3.65 STSTypeExporterListener Class Reference

Interface to be implemented to receive TypeExporterEvent events from a TypeExporter.

Inheritance diagram for STSTypeExporterListener:

```
+------------------+
| NSObject         |
|                  |
| STSTypeExporterListener |
```

Collaboration diagram for STSTypeExporterListener:

```
+------------------+
| NSObject         |
|                  |
| STSTypeExporterListener |
```
Instance Methods

- (void) - notifyEvent:

  Method to be notified from a TypeExporter.

3.65.1 Detailed Description

Interface to be implemented to receive TypeExporterEvent events from a TypeExporter.

Check out the 'Data export' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com

3.65.2 Method Documentation

3.65.2.1 - (void) notifyEvent: (STSTypeExporterEvent *) tee

Method to be notified from a TypeExporter.

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tee</td>
<td>[in]</td>
<td>Notified event.</td>
</tr>
</tbody>
</table>

The documentation for this class was generated from the following file:

- Sparksee.h

3.66 STSTypeList Class Reference

Sparksee type identifier list.

Inheritance diagram for STSTypeList:

```
   NSObject
   |
   |
  STSTypeList
```
Collaboration diagram for STSTypeList:

```
       NSObject
       
                          STSTypeList

Instance Methods

• (int) - count
  Number of elements in the list.
• (id) - init
  Constructor.
• (void) - add:
  Adds a Sparksee type identifier at the end of the list.
• (void) - clear
  Clears the list.
• (id) - initWithArray:
  Creates a new TypeList instance from the given array.
• (id) - initWithNSEnumerator:
  Creates a new TypeList instance from the given NSEnumerator.
• (STSTypeListIterator ∗) - iterator
  Gets a new TypeListIterator.

3.66.1 Detailed Description

Sparksee type identifier list.
It stores a Sparksee node or edge type identifier list.
Use TypeListIterator to access all elements into this collection.

Author

Sparsity Technologies http://www.sparsity-technologies.com

3.66.2 Method Documentation

3.66.2.1 - (void) add: (int) type

Adds a Sparksee type identifier at the end of the list.

Parameters

| type | [in] Sparksee type identifier. |
3.66.2.2 - (int) count

Number of elements in the list.

Returns

   Number of elements in the list.

3.66.2.3 - (id) init

Constructor.

This creates an empty list.

The documentation for this class was generated from the following file:

   • Sparksee.h

3.67 STSTypeListIterator Class Reference

TypeList iterator class.

Inheritance diagram for STSTypeListIterator:
Collaboration diagram for STSTypeListIterator:

```
NSObject

STSTypeList

theParent

STSTypeListIterator
```

Instance Methods

- (int) - next
  
  Moves to the next element.

- (BOOL) - hasNext
  
  Gets if there are more elements.

3.67.1 Detailed Description

TypeList iterator class.

Iterator to traverse all the Sparksee node or edge type identifiers into a TypeList instance.

Author

Sparsity Technologies [http://www.sparsity-technologies.com](http://www.sparsity-technologies.com)

3.67.2 Method Documentation

3.67.2.1 - (BOOL) hasNext

Gets if there are more elements.

Returns

TRUE if there are more elements, FALSE otherwise.

3.67.2.2 - (int) next

Moves to the next element.
Returns
The next element.

The documentation for this class was generated from the following file:

- Sparksee.h

3.68 STSTypeLoader Class Reference

Base TypeLoader class.

Inheritance diagram for STSTypeLoader:

![Inheritance Diagram]

Collaboration diagram for STSTypeLoader:

![Collaboration Diagram]

Instance Methods

- (void) setLogError:
  Sets a log error file.
- (void) setLogOff
Truns off all the error reporting.

- (void) - registerListener:
  Registers a new listener.
- (void) - run
  Run the loader.
- (void) - runTwoPhases
  Run the loader for two phases loading.
- (void) - runNPhases:
  Run the loader for N phases loading.
- (void) - setRowReader:
  Sets the input data source.
- (void) - setGraph:
  Sets the graph where the data will be loaded.
- (void) - setLocale:
  Sets the locale that will be used to read the data.
- (void) - setType:
  Sets the type to be loaded.
- (void) - setAttributes:
  Sets the list of Attributes.
- (void) - setAttributePositions:
  Sets the list of attribute positions.
- (void) - setTimestampFormat:
  Sets a specific timestamp format.
- (void) - setFrequency:
  Sets the frequency of listener notification.

3.68.1 Detailed Description

Base TypeLoader class.

Base class to load a node or edge type from a graph using a RowReader.
TypeLoaderListener can be registered to receive information about the progress of the load process by means of TypeLoaderEvent. The default frequency of notification to listeners is 100000.

Check out the 'Data import' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies [http://www.sparsity-technologies.com](http://www.sparsity-technologies.com)

3.68.2 Method Documentation

3.68.2.1 - (void) registerListener: (STSTypeLoaderListener ->) tel

Registers a new listener.

Parameters

| tel | TypeLoaderListener to be registered. |

3.68.2.2 - (void) run

Run the loader.
### Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.IO.IOException</td>
<td>null</td>
</tr>
<tr>
<td>System.ApplicationException</td>
<td>null</td>
</tr>
</tbody>
</table>

Implemented in `STSEdgeTypeLoader`, and `STSNodeTypeLoader`.

3.68.2.3 - (void) runNPhases: (int) `partitions`  

Run the loader for N phases loading.

Firstly load all objects (and create them if necessary) and secondly loads all the attributes. But in this case, attributes are loaded one by one. This way, if there are three attributes, then 4 traverses are necessary.

Working on this mode it is necessary to build a temporary file.

**Parameters**

| `partitions` | [in] Number of horizontal partitions to perform the load. |

3.68.2.4 - (void) runTwoPhases  

Run the loader for two phases loading.

Firstly load all objects (and create them if necessary) and secondly loads all the attributes.

Working on this mode it is necessary to build a temporary file.

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.IO.IOException</td>
<td>null</td>
</tr>
<tr>
<td>System.ApplicationException</td>
<td>null</td>
</tr>
</tbody>
</table>

Implemented in `STSEdgeTypeLoader`, and `STSNodeTypeLoader`.

3.68.2.5 - (void) setAttributePositions: (`STSInt32List` *) `attrsPos`  

Sets the list of attribute positions.

**Parameters**

| `attrsPos` | [in] Attribute positions (column index >=0). |

3.68.2.6 - (void) setAttributes: (`STSAttributeList` *) `attrs`  

Sets the list of Attributes.

**Parameters**

| `attrs` | [in] Attribute identifiers to be loaded |
3.68.2.7 - (void) setFrequency: (int) freq
Sets the frequency of listener notification.

Parameters

| freq | [in] Frequency in number of rows managed to notify progress to all listeners |

3.68.2.8 - (void) setGraph: (STSGraph *) graph
Sets the graph where the data will be loaded.

Parameters

| graph | [in] Graph |

3.68.2.9 - (void) setLocale: (NSString *) localeStr
Sets the locale that will be used to read the data.
It should match the locale used in the rowreader.

Parameters

| localeStr | [in] The locale string for the read data. See CSVReader. |

3.68.2.10 - (void) setLogError: (NSString *) path
Sets a log error file.
By default errors are thrown as a exception and the load process ends. If a log file is set, errors are logged there and the load process does not stop.

Parameters

| path | [in] The path to the error log file. |

Exceptions

| System.IO.IOException | If bad things happen opening the file. |

3.68.2.11 - (void) setLogOff
Turns off all the error reporting.
The log file will not be created and no exceptions for invalid data will be thrown. If you just want to turn off the logs, but abort at the first error what you should do is not call this method and not set a logError file.

3.68.2.12 - (void) setRowReader: (STSRowReader *) rr
Sets the input data source.

Parameters

| rr | [in] Input RowReader. |

3.68.2.13 - (void) setTimestampFormat: (NSString *) timestampFormat
Sets a specific timestamp format.
### Parameters

| timestamp-Format | [in] A string with the timestamp format definition. |

#### 3.68.2.14 - (void) setType: (int) type

Sets the type to be loaded.

| type | [in] Type identifier. |

The documentation for this class was generated from the following file:

- Sparksee.h

### 3.69 STSTypeLoaderEvent Class Reference

Provides information about the progress of a TypeLoader instance.

Inheritance diagram for STSTypeLoaderEvent:

```
NSObject

STSTypeLoaderEvent
```

Collaboration diagram for STSTypeLoaderEvent:

```
NSObject

STSTypeLoaderEvent
```
### Instance Methods

- (int) - `getTypId`
  
  *Gets the type identifier.*

- (long long) - `getCount`
  
  *Gets the current number of objects created.*

- (int) - `getPhase`
  
  *Gets the current phase.*

- (int) - `getTotalPhases`
  
  *Gets the total number of phases.*

- (int) - `getPartition`
  
  *Gets the current partition.*

- (int) - `getTotalPartitions`
  
  *Gets the total number of partitions.*

- (int) - `getTotalPartitionSteps`
  
  *Gets the total number of steps in the current partition.*

- (BOOL) - `isLast`
  
  *Gets if this is the last event or not.*

### 3.69.1 Detailed Description

Provides information about the progress of a TypeLoader instance.

Check out the 'Data import' section in the SPARKSEE User Manual for more details on this.

**Author**

Sparsity Technologies [http://www.sparsity-technologies.com](http://www.sparsity-technologies.com)

### 3.69.2 Method Documentation

#### 3.69.2.1 - (long long) `getCount`

*Gets the current number of objects created.*

**Returns**

The current number of objects created.

#### 3.69.2.2 - (int) `getPartition`

*Gets the current partition.*

**Returns**

The current partition.

#### 3.69.2.3 - (int) `getPhase`

*Gets the current phase.*

**Returns**

The current phase.
3.69.2.4 - (int) getTotalPartitions

Gets the total number of partitions.

Returns

The total number of partitions.

3.69.2.5 - (int) getTotalPartitionSteps

Gets the total number of steps in the current partition.

Returns

The total number steps in the current partition.

3.69.2.6 - (int) getTotalPhases

Gets the total number of phases.

Returns

The total number of phases.

3.69.2.7 - (int) getTypeId

Gets the type identifier.

Returns

The type identifier.

3.69.2.8 - (BOOL) isLast

Gets if this is the last event or not.

Returns

TRUE if this is the last event, FALSE otherwise.

The documentation for this class was generated from the following file:

- Sparksee.h

3.70 STSTypeLoaderListener Class Reference

Interface to be implemented to receive TypeLoaderEvent events from a TypeLoader.
Inheritance diagram for STSTypeLoaderListener:

```
NSObject

STSTypeLoaderListener
```

Collaboration diagram for STSTypeLoaderListener:

```
NSObject

STSTypeLoaderListener
```

Instance Methods

- (void) - notifyEvent:
  
  Method to receive events from a Loader.

3.70.1 Detailed Description

Interface to be implemented to receive TypeLoaderEvent events from a TypeLoader.
Check out the 'Data import' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies [http://www.sparsity-technologies.com](http://www.sparsity-technologies.com)

3.70.2 Method Documentation

3.70.2.1 - (void) notifyEvent: (STSTypeLoaderEvent *) ev

Method to receive events from a Loader.
Parameters

| ev       | Loader::LoaderEvent with information from a running Loader. |

The documentation for this class was generated from the following file:

- Sparksee.h

3.71 STSValue Class Reference

Value class.

Inheritance diagram for STSValue:

![Inheritance Diagram](image)

Collaboration diagram for STSValue:

![Collaboration Diagram](image)

Instance Methods

- (id) - init

  Creates a new instance.

- (id) - initWithValue:

  Copy constructor.

- (BOOL) - isNull

  Gets if this is a NULL Value.

- (void) - setNullVoid
Sets the Value to NULL.

- (enum STSDataType) - getDataType
  Gets the DataType.

- (BOOL) - getBoolean
  Gets Boolean Value.

- (int) - getInteger
  Gets Integer Value.

- (long long) - getLong
  Gets Long Value.

- (double) - getDouble
  Gets Double Value.

- (long long) - getTimestamp
  Gets Timestamp Value.

- (NSString *) - getString
  Gets String Value.

- (long long) - getOid
  Gets OID Value.

- (void) - setBooleanVoid:
  Sets the Value.

- (void) - setIntegerVoid:
  Sets the Value.

- (void) - setLongVoid:
  Sets the Value.

- (void) - setDoubleVoid:
  Sets the Value.

- (void) - setTimestampVoid:
  Sets the Value.

- (void) - setTimestampVoidWithYear:month:day:hour:minutes:seconds:millisecs:
  Sets the Value.

- (void) - setStringVoid:
  Sets the Value.

- (void) - setOidVoid:
  Sets the OID Value.

- (void) - setVoid:
  Sets the Value.

- (int) - compare:
  Compares with the given Value.

- (BOOL) - equals:
  Compares with the given Value.

- (NSString *) - stringValue
  Returns the receiver's value as a human-readable string.

- (BOOL) - isEqual:
  Check if both Value instances are equal.

- (NSUInteger) - hash
  Get the hash value of this Value.

- (STSValue *) - setNull
  Sets the value to NULL.

- (STSValue *) - setBoolean:
  Sets the value to the given boolean.

- (STSValue *) - setInteger:
  Sets the value to the given integer number.
• (STSValue *) - setLong:
  Sets the value to the given long long number.
• (STSValue *) - setDouble:
  Sets the value to the given double number.
• (STSValue *) - setTimestamp:
  Sets the value to the given timestamp.
• (STSValue *) - setTimestampWithYear:month:day:hour:minutes:seconds:milliseconds:
  Sets the value to the given timestamp.
• (STSValue *) - setString:
  Sets the value to the given string.
• (STSValue *) - setOid:
  Sets the value to the given OID.
• (STSValue *) - set:
  Sets the value to the given Value.
• (NSDate *) - getTimestampAsNSDate
  Gets Timestamp Value as a NSDate.
• (STSValue *) - setTimestampWithNSDate:
  Sets the value to the given NSDate timestamp.

Class Methods

• (int) + getMaxLengthString
  Maximum number of characters allowed for a String.

3.71.1 Detailed Description

Value class.
It is a container which stores a value and its data type (domain). A Value can be NULL.

Author
  Sparsity Technologies http://www.sparsity-technologies.com

3.71.2 Method Documentation

3.71.2.1 - (int) compare: (STSValue *) value

Compares with the given Value.
It does not work if the given Value objects does not have the same DataType.

Parameters

| value | Given value to compare to. |

Returns

  0 if this Value is equal to the given one; a value less than 0 if this Value is less than the given one; and a value greater than 0 if this Value is greater than the given one.

3.71.2.2 - (BOOL) equals: (STSValue *) value

Compares with the given Value.
It does not work if the given Value objects does not have the same DataType.
3.71 STSValue Class Reference

Parameters

| value | Given value to compare to. |

Returns

TRUE if this Value is equal to the given one; FALSE otherwise.

3.71.2.3 - (BOOL) getBoolean

Gets Boolean Value.
This must be a non-NULL Boolean Value.

Returns

The Boolean Value.

3.71.2.4 - (enum STSDataType) getDataType

Gets the DataType.
Value cannot be NULL.

Returns

The DataType.

3.71.2.5 - (double) getDouble

Gets Double Value.
This must be a non-NULL Double Value.

Returns

The Double Value.

3.71.2.6 - (int) getInteger

Gets Integer Value.
This must be a non-NULL Integer Value.

Returns

The Integer Value.

3.71.2.7 - (long long) getLong

Gets Long Value.
This must be a non-NULL Long Value.

Returns

The Long Value.
3.71.2.8 - (long long) getOid

Gets OID Value.
This must be a non-NULL OID Value.

Returns
The OID Value.

3.71.2.9 - (NSString *) getString

Gets String Value.
This must be a non-NULL String Value.

Returns
The String Value.

3.71.2.10 - (long long) getTimestamp

Gets Timestamp Value.
This must be a non-NULL Timestamp Value.

Returns
The Timestamp Value.

3.71.2.11 - (NSDate *) getTimestampAsNSDate

Gets Timestamp Value as a NSDate.
This must be a non-NULL Timestamp Value.

Returns
The Timestamp Value.

3.71.2.12 - (id) init

Creates a new instance.
It creates a NULL Value.

3.71.2.13 - (id) initWithValue: (STSValue *) value

Copy constructor.

Parameters

| value | [in] Value to be copied. |

3.71.2.14 - (BOOL) isNull

Gets if this is a NULL Value.

Returns
TRUE if this is a NULL Value, FALSE otherwise.
3.71.2.15 - (STSValue*) set: (STSValue*) value

Sets the value to the given Value.

Parameters


Returns

Returns this Value.

3.71.2.16 - (STSValue*) setBoolean: (BOOL) value

Sets the value to the given boolean.

Parameters


Returns

Returns this Value.

3.71.2.17 - (void) setBooleanVoid: (BOOL) value

Sets the Value.

Parameters

value [in] New Boolean value.

3.71.2.18 - (STSValue*) setDouble: (double) value

Sets the value to the given double number.

Parameters


Returns

Returns this Value.

3.71.2.19 - (void) setDoubleVoid: (double) value

Sets the Value.

Parameters

value [in] New Double value.

3.71.2.20 - (STSValue*) setInteger: (int) value

Sets the value to the given integer number.
3.71.2.21 - (void) setIntegerVoid: (int) value
Sets the Value.
Parameters

| value | [in] New Integer value. |

3.71.2.22 - (STSValue*) setLong: (long long) value
Sets the value to the given long long number.
Parameters


3.71.2.23 - (void) setLongVoid: (long long) value
Sets the Value.
Parameters


3.71.2.24 - (STSValue*) setNull
Sets the value to NULL.
Returns

| Returns this Value. |

3.71.2.25 - (STSValue*) setOid: (long long) value
Sets the value to the given OID.
Parameters


3.71.2.26 - (void) setOidVoid: (long long) value
Sets the OID Value.
Parameters

value [in] New OID value.

3.71.2.27 - (STSValue *) setString: (NSString *) value

Sets the value to the given string.

Parameters


Returns

Returns this Value.

3.71.2.28 - (void) setStringValue: (NSString *) value

Sets the Value.

Parameters

value [in] New String value.

3.71.2.29 - (STSValue *) setTimestamp: (long long) value

Sets the value to the given timestamp.

Parameters


Returns

Returns this Value.

3.71.2.30 - (void) setTimestampVoid: (long long) value

Sets the Value.

Parameters


3.71.2.31 - (void) setTimestampVoidWithYear: (int) year month:(int) month day:(int) day hour:(int) hour minutes:(int) minutes seconds:(int) seconds millisecs:(int) millisecs

Sets the Value.

Parameters

month [in] The month ([1..12]).
day [in] The of the month ([1..31]).
hour [in] The hour ([0..23]).
minutes [in] The minutes ([0..59]).
seconds [in] The seconds ([0..59]).
millisecs [in] The milliseconds ([0..999]).
3.71.2.32  - (STSValue *) setTimestampWithNSDate: (NSDate *) date

Sets the value to the given NSDate timestamp.

Parameters


Returns

Returns this Value.

3.71.2.33  - (STSValue *) setTimestampWithYear: (int) year month:(int) month day:(int) day hour:(int) hour minutes:(int) minutes seconds:(int) seconds milliseconds:(int) millisecs

Sets the value to the given timestamp.

Parameters

| month    | [in] The month ([1..12]). |
| day      | [in] The day of the month ([1..31]). |
| hour     | [in] The hour ([0..23]). |
| minutes  | [in] The minutes ([0..59]). |
| seconds  | [in] The seconds ([0..59]). |
| millisecs| [in] The milliseconds ([0..999]). |

Returns

Returns this Value.

3.71.2.34  - (void) setVoid: (STSValue *) value

Sets the Value.

Parameters

| value     | [in] New value. |

The documentation for this class was generated from the following file:

- Sparksee.h
Inheritance diagram for STSValueList:

Collaboration diagram for STSValueList:

Instance Methods

- (int) - count
  Number of elements in the list.
- (id) - init
  Constructor.
- (void) - clear
  Clears the list.
- (void) - add:
  Adds a value to the end of the list.
- (STSValue *) - get:
  Returns the Value at the specified position in the list.
- (id) - initWithArray:
  Creates a new ValueList instance from the given array.
- (id) - initWithNSEnumerator:
  Creates a new ValueList instance from the given NSEnumerator.
- (STSValueListIterator *) - iterator
  Gets a new ValueListIterator.
3.72.1 Detailed Description

Value list.
It stores a Value list.
Use ValueListIterator to access all elements into this collection.

Author

Sparsity Technologies [http://www.sparsity-technologies.com](http://www.sparsity-technologies.com)

3.72.2 Method Documentation

3.72.2.1 - (void) add: (STSValue *) value

Adds a value to the end of the list.

Parameters

| value | [in] The value to add |

3.72.2.2 - (int) count

Number of elements in the list.

Returns

Number of elements in the list.

3.72.2.3 - (STSValue *) get: (int) index

Returns the Value at the specified position in the list.

Parameters

| index | [in] Index of the element to return, starting at 0. |

3.72.2.4 - (id) init

Constructor.
This creates an empty list.

The documentation for this class was generated from the following file:

- Sparksee.h

3.73 STSValueListIterator Class Reference

ValueList iterator class.
Inheritance diagram for STSValueListIterator:

Collaboration diagram for STSValueListIterator:

Instance Methods

- (STSValue *) next
  Moves to the next element.
- (BOOL) hasNext
  Gets if there are more elements.

3.73.1 Detailed Description

ValueList iterator class.
Iterator to traverse all the values into a ValueList instance.
3.74 STSValues Class Reference

Value set class.

Inheritance diagram for STSValues:

```
  NSObject
    ↓
  STSValues
```
Collaboration diagram for STSValues:

![Collaboration Diagram]

### Instance Methods

- **(long long) - count**
  
  *Gets the number of elements into the collection.*

- **(STSValuesIterator *) - iterator:**
  
  *Gets a ValuesIterator.*

- **(void) - close**
  
  *Closes the Values instance.*

- **(BOOL) - isClosed**
  
  *Check if the Values instance is closed.*

### 3.74.1 Detailed Description

Value set class.

This is a set of Value instances, that is there is no duplicated elements.

Use a ValuesIterator to traverse all the elements into the set.

When the Values instance is closed, it closes all existing and non-closed ValuesIterator instances too.

**Author**

Sparsity Technologies [http://www.sparsity-technologies.com](http://www.sparsity-technologies.com)

### 3.74.2 Method Documentation

#### 3.74.2.1 - (void) close

Closes the Values instance.

It must be called to ensure the integrity of all data.

#### 3.74.2.2 - (long long) count

Gets the number of elements into the collection.

Returns

The number of elements into the collection.
3.74.2.3 - (STSVvaluesIteration+) iterator: (enum STSOrder) order

Gets a ValuesIterator.

Parameters

| order | [in] Ascending or descending order.

Returns

ValuesIterator instance.

The documentation for this class was generated from the following file:

- Sparksee.h

3.75 STSVvaluesIteration Class Reference

Values iterator class.

Inheritance diagram for STSVvaluesIteration:

- NSObject

Collaboration diagram for STSVvaluesIteration:

- NSObject

Instance Methods

- (BOOL) - hasNext
3.75 Detailed Description

Values iterator class.
It allows for traversing all the elements into a Values instance.

Author
Sparsity Technologies [http://www.sparsity-technologies.com](http://www.sparsity-technologies.com)

3.75.2 Method Documentation

3.75.2.1 - (void) close
Closes the ValuesIterator instance.
It must be called to ensure the integrity of all data.

3.75.2.2 - (BOOL) hasNext
Gets if there are more elements to traverse.

Returns
TRUE if there are more elements to traverse, FALSE otherwise.

3.75.2.3 - (STSValue *) next
Gets the next element to traverse.

Returns
The next element.

The documentation for this class was generated from the following file:

• Sparksee.h

3.76 STSWeakConnectivity Class Reference

WeakConnectivity class.
Instance Methods

- (void) - addEdgeType:
  Allows connectivity through edges of the given type.
- (void) - addAllEdgeTypes
  Allows connectivity through all edge types of the graph.
3.76 STSWeakConnectivity Class Reference

3.76.1 Detailed Description

WeakConnectivity class.

Any class implementing this abstract class can be used to solve the problem of finding weakly connected components in an undirected graph or in a directed graph which will be considered as an undirected one.

It consists in finding components where every pair \((u,v)\) of nodes contained in it has a path from \(u\) to \(v\) and from \(v\) to \(u\).

It is possible to set some restrictions after constructing a new instance of this class and before running it in order to limit the results.

After the execution, we can retrieve the results stored in an instance of the ConnectedComponents class using the getConnectedComponents method.

Check out the 'Algorithms' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com

3.76.2 Method Documentation

3.76.2.1 - (void) addAllEdgeTypes

Allows connectivity through all edge types of the graph.

In a weak connectivity the edges can be used in Any direction.

3.76.2.2 - (void) addEdgeType: (int) type

Allows connectivity through edges of the given type.

In a weak connectivity the edges can be used in Any direction.

Parameters

| type | [in] Edge type. |
3.76.2.3 - (void) addNodeType: (int) t

Allows connectivity through nodes of the given type.

Parameters

| t  | null |

3.76.2.4 - (void) close

Closes the Connectivity instance.
It must be called to ensure the integrity of all data.

3.76.2.5 - (void) excludeEdges: (STSObjects *) edges

Set which edges can't be used.
This will replace any previously specified set of excluded edges. Should only be used to exclude the usage of specific edges from allowed edge types because it's less efficient than not allowing an edge type.

Parameters

| edges | [in] A set of edge identifiers that must be kept intact until the destruction of the class. |

3.76.2.6 - (void) excludeNodes: (STSObjects *) nodes

Set which nodes can't be used.
This will replace any previously specified set of excluded nodes. Should only be used to exclude the usage of specific nodes from allowed node types because it's less efficient than not allowing a node type.

Parameters

| nodes | [in] A set of node identifiers that must be kept intact until the destruction of the class. |

3.76.2.7 - (STSConnectedComponents *) getConnectedComponents

Returns the results generated by the execution of the algorithm.
These results contain information related to the connected components found as the number of different components, the set of nodes contained in each component or many other data.

Returns

Returns an instance of the class ConnectedComponents which contain information related to the connected components found.

3.76.2.8 - (void) run

Runs the algorithm in order to find the connected components.
This method can be called only once.
Implemented in STSStrongConnectivityGabow, and STSWeakConnectivityDFS.

3.76.2.9 - (void) setMaterializedAttribute: (NSString *) attributeName

Creates a new common attribute type for all node types in the graph in order to store, persistently, the results related to the connected components found while executing this algorithm.
Whenever the user wants to retrieve the results, even when the graph has been closed and opened again, it is only necessary to create a new instance of the class ConnectedComponents indicating the graph and the name of the
common attribute type which stores the results. This instance will have all the information related to the connected components found in the moment of the execution of the algorithm that stored this data.

It is possible to run the algorithm without specifying this parameter in order to avoid materializing the results of the execution.

Parameters

| attributeName | [in] The name of the common attribute type for all node types in the graph which will store persistently the results generated by the execution of the algorithm. |

The documentation for this class was generated from the following file:

- Sparksee.h

3.77 STSWeakConnectivityDFS Class Reference

WeakConnectivityDFS class.

Inheritance diagram for STSWeakConnectivityDFS:
Collaboration diagram for STSWeakConnectivityDFS:

- **NSObject**
- **STSConnectivity**
- **STSWeakConnectivity**
- **STSWeakConnectivityDFS**

### Instance Methods

- **(id)** - initWithSession:
  
  Creates a new instance of WeakConnectivityDFS.

- **(void)** - run
  
  Executes the algorithm.

- **(void)** - addEdgeType:
  
  Allows connectivity through edges of the given type.

- **(void)** - addAllEdgeTypes
  
  Allows connectivity through all edge types of the graph.

- **(void)** - addNodeType:
  
  Allows connectivity through nodes of the given type.

- **(void)** - addAllNodeTypes
  
  Allows connectivity through all node types of the graph.

- **(void)** - excludeNodes:
  
  Set which nodes can’t be used.

- **(void)** - excludeEdges:
  
  Set which edges can’t be used.

- **(STSConnectedComponents ∗)** - getConnectedComponents
  
  Returns the results generated by the execution of the algorithm.

- **(void)** - setMaterializedAttribute:
  
  Creates a new common attribute type for all node types in the graph in order to store, persistently, the results related to the connected components found while executing this algorithm.

- **(void)** - close
  
  Closes the Connectivity instance.

- **(BOOL)** - isClosed
  
  Check if the Connectivity instance is closed.
3.77.1 Detailed Description

WeakConnectivityDFS class.

This class can be used to solve the problem of finding weakly connected components in an undirected graph or in a directed graph which will be considered as an undirected one.

It consists in finding components where every pair \((u,v)\) of nodes contained in it has a path from \(u\) to \(v\) and from \(v\) to \(u\). This implementation is based on the Depth-First Search (DFS) algorithm.

It is possible to set some restrictions after constructing a new instance of this class and before running it in order to limit the results.

After the execution, we can retrieve the results stored in an instance of the ConnectedComponents class using the getConnectedComponents method.

Check out the 'Algorithms' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com

3.77.2 Method Documentation

3.77.2.1 - (void) addAllEdgeTypes

Allows connectivity through all edge types of the graph.

In a weak connectivity the edges can be used in Any direction.

3.77.2.2 - (void) addEdgeType: (int) type

Allows connectivity through edges of the given type.

In a weak connectivity the edges can be used in Any direction.

Parameters

| type | [in] Edge type. |

3.77.2.3 - (void) addNodeType: (int) t

Allows connectivity through nodes of the given type.

Parameters

| t | null |

3.77.2.4 - (void) close

Closes the Connectivity instance.

It must be called to ensure the integrity of all data.

3.77.2.5 - (void) excludeEdges: (STSOBJECTS *) edges

Set which edges can't be used.

This will replace any previously specified set of excluded edges. Should only be used to exclude the usage of specific edges from allowed edge types because it's less efficient than not allowing an edge type.
Parameters

| edges | [in] A set of edge identifiers that must be kept intact until the destruction of the class. |

3.77.2.6 - (void) excludeNodes: (STSObjects *) nodes

Set which nodes can't be used.
This will replace any previously specified set of excluded nodes. Should only be used to exclude the usage of specific nodes from allowed node types because it's less efficient than not allowing a node type.

Parameters

| nodes | [in] A set of node identifiers that must be kept intact until the destruction of the class. |

3.77.2.7 - (STSConnectedComponents *) getConnectedComponents

Returns the results generated by the execution of the algorithm.
These results contain information related to the connected components found as the number of different components, the set of nodes contained in each component or many other data.

Returns

Returns an instance of the class ConnectedComponents which contain information related to the connected components found.

3.77.2.8 - (id) initWithSession: (STSSession *) session

Creates a new instance of WeakConnectivityDFS.
After creating this instance is required to indicate the set of edge types and the set of node types which will be navigated through while traversing the graph in order to find the weak connected components.

Parameters

| session | [in] Session to get the graph from and calculate the connectivity |

3.77.2.9 - (void) setMaterializedAttribute: (NSString *) attributeName

Creates a new common attribute type for all node types in the graph in order to store, persistently, the results related to the connected components found while executing this algorithm.

Whenever the user wants to retrieve the results, even when the graph has been closed and opened again, it is only necessary to create a new instance of the class ConnectedComponents indicating the graph and the name of the common attribute type which stores the results. This instance will have all the information related to the connected components found in the moment of the execution of the algorithm that stored this data.

It is possible to run the algorithm without specifying this parameter in order to avoid materializing the results of the execution.

Parameters

| attributeName | [in] The name of the common attribute type for all node types in the graph which will store persistently the results generated by the execution of the algorithm. |

The documentation for this class was generated from the following file:

- Sparksee.h
Inheritance diagram for SWIGTYPE_p_wchar_t:

Collaboration diagram for SWIGTYPE_p_wchar_t:

The documentation for this class was generated from the following file:

- Sparksee.h
Index

add:
- STSAttributeList, 11
- STSBooleanList, 18
- STSInt32List, 98
- STSOidList, 115
- STSTypeList, 212
- STSValueList, 233

addAllEdgeTypes:
- STSCommunitiesSCD, 22
- STSDisjointCommunityDetection, 56
- STSWeakConnectivity, 240
- STSWeakConnectivityDFS, 244

addAllEdgeTypes:
- STSContext, 33
- STSShortestPath, 149
- STSSinglePairShortestPath, 152
- STSSinglePairShortestPathBFS, 156
- STSSinglePairShortestPathDijkstra, 160
- STSTraversal, 194
- STSTraversalBFS, 197
- STSTraversalDFS, 201

addEdgeType:
- STSCommunitiesSCD, 22
- STSDisjointCommunityDetection, 56
- STSWeakConnectivity, 240
- STSWeakConnectivityDFS, 244

addEdgeType:d:
- STSContext, 33

addEdgeType:dir:
- STSShortestPath, 149
- STSSinglePairShortestPath, 152
- STSSinglePairShortestPathBFS, 156
- STSSinglePairShortestPathDijkstra, 160
- STSTraversal, 194
- STSTraversalBFS, 197
- STSTraversalDFS, 201

addNodeType:
- STSCommunitiesSCD, 22
- STSDisjointCommunityDetection, 56
- STSWeakConnectivity, 240
- STSWeakConnectivityDFS, 244

addWeightedEdgeType:dir:attr:
- STSSinglePairShortestPathDijkstra, 160

backup:
- STSSession, 147
- STSStraversalBFS, 198
- STSStraversalDFS, 201

checkOnlyExistence:
- STSSinglePairShortestPathBFS, 156
- STSSession, 147

clone:
- STSObjects, 115

cloneWithObjects:
- STSObjects, 115

close:
- STSCommunitiesSCD, 22
- STSConnectivity, 31
- STSShortestPath, 149
- STSSinglePairShortestPathBFS, 156
- STSSinglePairShortestPathDijkstra, 160
- STSStraversal, 194
- STSStraversalBFS, 197
- STSStraversalDFS, 201

combineDifference:objs2:
- STSObjects, 115

combineIntersection:objs2:
- STSObjects, 116

combineUnion:objs2:
INDEX

STSObjects, 116
compare:
STSValue, 225
compute

STSContext, 34
computeWithArguments:node:nodeTypes:edgeTypes:dir:maxhops:include:

STSContext, 34
contains:

STSObjects, 116
count

STSAttributeList, 11
STSBinaryList, 18
STSSInt32List, 98
STSObjects, 117
STSOidList, 122
STSStructSetList, 137
STSStringList, 180
STSTypeList, 212
STSStructList, 233
STSVariantList, 236
countEdges

STSSpark, 84
countNodes

STSSpark, 84
create:alias:

STSSpark, 164
createAttribute:name:dt:kind:

STSSpark, 84
createAttributeWithDefault:name:dt:kind:defaultValue:

STSSpark, 84
createEdge:tail:head:

STSSpark, 84
createEdgeType:directed:neighbors:

STSSpark, 84
createEdgeWithAttributes:tailAttr:tailV:headAttr:headV:

STSSpark, 84
createNode:

STSSpark, 84
createNodeType:

STSSpark, 84
createObjects

STSSession, 147
createRestrictedEdgeType:tail:head:neighbors:

STSSpark, 83
createSessionAttribute:dt:kind:

STSSpark, 83
createSessionAttributeWithDefault:dt:kind:defaultValue:

STSSpark, 83
degree:etype:dir:

STSSpark, 84
difference:

STSObjects, 117
drop:

STSSpark, 84
dropWithObjects:

STSSpark, 84
dumpData:

STSSpark, 84
dumpStorage:

STSSpark, 84
dumpStorage:

STSSpark, 84
edges:tail:head:

STSSpark, 85
enableType:

STSDefaultExport, 50
STSExportManager, 75
equals:

STSObjects, 117
STSVariantList, 225
excludeEdges:

STSCommunitiesSCD, 22
STSClusterDetection, 25
STSClustered, 31
STSContext, 34
STSDisjointCommunityDetection, 57
STSShortestPath, 149
STSSingleShortestPath, 153
STSSinglePairShortestPathBFS, 157
STSSinglePairShortestPathDijkstra, 161
STSStructuredCommunityDetection, 185
STSStructuredConnectivity, 189
STSStructuredTraversal, 195
STSStructuredTraversalBFS, 198
STSStructuredTraversalDFS, 201
STSStructuredWeakConnectivity, 241
STSStructuredWeakConnectivityDFS, 244
excludeNodes:

STSCommunitiesSCD, 23
STSClusterDetection, 25
STSClustered, 31
STSContext, 35
STSDisjointCommunityDetection, 57
STSShortestPath, 150
STSSingleShortestPath, 153
STSSinglePairShortestPathBFS, 157
STSSinglePairShortestPathDijkstra, 161
STSStructuredCommunityDetection, 185
STSStructuredConnectivity, 189
STSStructuredTraversal, 195
STSStructuredTraversalBFS, 198
STSStructuredTraversalDFS, 201
STSStructuredWeakConnectivity, 241
STSStructuredWeakConnectivityDFS, 244
execute:

STSQuery, 129
exists:

STSObjects, 117
explode:etype:dir:

STSSpark, 85
explodeWithObjects:etype:dir:

STSSpark, 85
exportGraph:type:em:

STSSpark, 85
fetch:

STSQueryStream, 132

Generated on Tue Jun 2 2015 15:49:36 for Sparksee by Doxygen
findAttribute: name:
   STSGraph, 86
findAttributes:
   STSGraph, 86
findEdge: tail: head:
   STSGraph, 86
findEdgeTypes
   STSGraph, 86
findNodeTypes
   STSGraph, 87
findObject: value:
   STSGraph, 87
findOrCreateEdge: tail: head:
   STSGraph, 87
findOrCreateObject: value:
   STSGraph, 87
findType:
   STSGraph, 88
findTypes
   STSGraph, 88
fixCurrentCacheMaxSize
   STSDatabase, 45
generateSchemaScript: db:
   STSScriptParser, 144
get:
   STSResultSetList, 137
   STSValueList, 233
get: def:
   STSSparkseeProperties, 178
getAlias
   STSDatabase, 45
getAreNeighborsIndexed
   STSType, 204
getAttribute:
   STSGraph, 88
getAttributeInValue: attr: value:
   STSGraph, 88
getAttributeIntervalCount: lower: includeLower: higher: includeHigher:
   STSGraph, 88
getAttributeStatistics: basic:
   STSGraph, 89
getAttributeText: attr:
   STSGraph, 89
getAttributeWithOid: attr:
   STSGraph, 89
getAttributes:
   STSGraph, 89
getAvailableMem
   STSPlatformStatistics, 127
getAvgLengthString
   STSAttributeStatistics, 15
getBoolean
   STSValue, 226
getBoolean: def:
   STSSparkseeProperties, 178
getCache
   STSDatabaseStatistics, 47
getCacheMaxSize
   STSDatabase, 45
   STSSparkseeConfig, 169
getCacheStatisticsEnabled
   STSSparkseeConfig, 169
getCacheStatisticsFile
   STSSparkseeConfig, 169
getCacheStatisticsSnapshotTime
   STSSparkseeConfig, 169
getColorRGB
   STSEdgeExport, 62
   STSNodeExport, 102
ggetColorRed: green: blue: alpha:
   STSEdgeExport, 61
   STSNodeExport, 102
column:
   STSResultSet, 134
columnDataType:
   STSResultSet, 134
columnIndex:
   STSResultSet, 134
columnName:
   STSResultSet, 135
columnWithValue: value:
   STSResultSet, 135
columns:
   STSCommunitiesSCD, 23
   STSDisjointCommunityDetection, 57
community:
   STSDisjointCommunities, 53
connectedComponent:
   STSConnectedComponents, 28
connectedComponents:
   STSConnectivity, 31
   STSStrongConnectivity, 186
   STSStrongConnectivityGabow, 189
   STSWeakConnectivity, 241
   STSWeakConnectivityDFS, 245
cost:
   STSSinglePairShortestPath, 153
   STSSinglePairShortestPathBFS, 157
   STSSinglePairShortestPathDijkstra, 161
count:
   STSAttribute, 9
   STSConnectedComponents, 28
   STSDisjointCommunities, 53
   STSTypeExporterEvent, 209
   STSTypeLoaderEvent, 220
currentDepth:
   STSTraversal, 195
   STSTraversalBFS, 198
   STSTraversalDFS, 202
data:
   STSDatabaseStatistics, 47
dataType:
   STSAttribute, 9
   STSValue, 226
distinct
getPartition
   STSTypeLoaderEvent, 220

getPath
   STSDatabase, 45

getPathAsEdges
   STSSinglePairShortestPath, 153
   STSSinglePairShortestPathBFS, 157
   STSSinglePairShortestPathDijkstra, 161

getPathAsNodes
   STSSinglePairShortestPath, 153
   STSSinglePairShortestPathBFS, 157
   STSSinglePairShortestPathDijkstra, 161

getPhase
   STSTypeLoaderEvent, 220

getPoolClusterSize
   STSSparkseeConfig, 171

getPoolFrameSize
   STSSparkseeConfig, 171

getPoolPersistentMaxSize
   STSSparkseeConfig, 171

getPoolPersistentMinSize
   STSSparkseeConfig, 171

getPoolTemporaryMaxSize
   STSSparkseeConfig, 171

getPoolTemporaryMinSize
   STSSparkseeConfig, 172

getRow
   STSCSVReader, 38
   STSRowReader, 140

getSessions
   STSDatabaseStatistics, 48

getShape
   STSNodeExport, 103

getSize
   STSAttribute, 9

getSize:
   STSConnectedComponents, 28
   STSDisjointCommunities, 54

getStatistics:
   STSDatabase, 45

getRead
   STSDatabaseStatistics, 48

getRealTime
   STSPlatformStatistics, 127

getRecoveryCacheMaxSize
   STSSparkseeConfig, 172

getRecoveryCheckpointTime
   STSSparkseeConfig, 172

getRecoveryEnabled
   STSSparkseeConfig, 172

getRecoveryLogFile
   STSSparkseeConfig, 172

getRestrictedFrom
   STSType, 205

getRestrictedTo
   STSType, 205

getRollbackEnabled
   STSSparkseeConfig, 172

getRow
   STSCSVReader, 38
   STSRowReader, 140

getSystemTime
   STSPlatformStatistics, 127

getTail
   STSEdgeData, 59

getTemp
   STSDatabaseStatistics, 48

getTimeUnit:def
   STSSparkseeProperties, 178

timestamp
   STSValue, 227

getTimestamp
   STSValue, 227

getTimestampAsNSDate
   STSValue, 227

getTotal
   STSAttributeStatistics, 16
   STSTypeExporterEvent, 209

getTotalMem
   STSPlatformStatistics, 127

getTotalPartitionSteps
   STSTypeLoaderEvent, 221

getTotalPartitions
   STSTypeLoaderEvent, 220

getTotalPhases
   STSTypeLoaderEvent, 221

getTotal
   STSAttributeStatistics, 16
   STSTypeExporterEvent, 209

getTotalMem
   STSPlatformStatistics, 127

getTotalPartitionSteps
   STSTypeLoaderEvent, 221

getTotalPartitions
   STSTypeLoaderEvent, 220

getTotalPhases
   STSTypeLoaderEvent, 221

type:
   STSGraph, 90

getTypeld
   STSAttribute, 9
   STSTypeExporterEvent, 209
   STSTypeLoaderEvent, 221

getUserTime
   STSPlatformStatistics, 128

getValues:
   STSGraph, 90

getVariance
   STSAttributeStatistics, 17

getWidth
   STSEdgeExport, 62
   STSNodeExport, 103

getWrite
   STSDatabaseStatistics, 48

hasNext
   STSAttributeListIterator, 13
   STSBooleanListIterator, 20
   STSInt32ListIterator, 99
   STSObjectListIterator, 120
   STSObjectIdListIterator, 124
   STSResultSetListIterator, 139
   STSStringListIterator, 182
   STSTraversal, 195
   STSTraversalBFS, 198
   STSTraversalDFS, 202
   STSTypeListIterator, 214
   STSValueListIterator, 235
   STSValuesListIterator, 238

heads:
**STSGraph**, 91

indexAttribute.kind:

- STSGraph, 91

**init**

- STSAttributeList, 11
- STSBooleanList, 18
- STSInt32List, 98
- STSOidList, 122
- STSResultSetList, 137
- STSSparkseeConfig, 172
- STSStringList, 181
- STSTypeList, 213
- STSValue, 227
- STSValueList, 233

**initWithAppend**

- STSTextStream, 192

**initWithConfig**

- STSSparksee, 164

**initWithNumInvalidOIDs**

- STSOidList, 122

**initWithRowReader**

- STSNodeTypeLoader, 110

**initWithRowWriter**

- STSNodeTypeExporter, 107

**initWithRowWriter**

- STSNodeTypeExporter, 66

**initWithS**

- STSConnectedComponents, 28

**initWithSession**

- STSCommunitiesSCD, 23
- STSStrongConnectivityGabow, 189
- STSWeakConnectivityDFS, 245

**initWithSession**

- STSDisjointCommunities, 54

**initWithSession**

- STSTextStream, 192

**isNull**

- STSValue, 227

**intersection**

- STSObjects, 117

**isLast**

- STSTypeExport, 103

**isLast**

- STSTypeExport, 210

**isLast**

- STSTypeExport, 221

**isNull**

- STSTextStream, 192

**isNull**

- STSValue, 227

**isSessionAttribute**

- STSAttribute, 9

**iterator**

- STSObjects, 118

**iterator**

- STSValues, 236

**iterator**

- STSObjects, 118

**iterator**

- STSObjects, 118

**load**

- STSSparkseeProperties, 179

**neighbors**

- STSGraph, 91

**neighbors**

- STSGraph, 91

**init**

- STSAttributeListIterator, 13

**init**

- STSBooleanListIterator, 20

**init**

- STSInt32ListIterator, 99

**init**

- STSOidListIterator, 124

**init**

- STSResultSetListIterator, 139

**init**

- STSStringListIterator, 182

**init**

- STSTypeListIterator, 214

**init**

- STSValueListIterator, 235

**init**

- STSValuesIterator, 238

**notifyEvent**

- STSTypeExporterListener, 211

**notifyEvent**

- STSExceptionListener, 222

**open**

- STSCSVReader, 38

**open**

- STSCSVWriter, 42

**open**

- STSSparksee, 164

**parse**

- STSScriptParser, 145

**prepare**

- STSDefaultExport, 52

**prepare**

- STSExportManager, 77

**prepare**

- STSQueryStream, 132

**read**

- STSCSVReader, 38

**read**

- STSRowReader, 141

**readString**

- STSTextStream, 192

**registerListener**

- STSEdgeTypeExporter, 66

**registerListener**

- STSExceptionListener, 70

**registerListener**

- STSDefaultExport, 107

**registerListener**

- STSExportManager, 110

**registerListener**

- STSExportManager, 207

**registerListener**

- STSExceptionListener, 216

**remove**

- STSObjects, 118
**_INDEX 254**

- **STSOObjects**, 118
  - **removeAttribute**: 92
  - **removeType**: 92
  - **renameAttribute**:newName: 92
  - **renameType**:newName: 92
  - **renameTypeWithName**:newName: 92
  - **reset**: 92
  - **STSCSVReader**, 38
  - **STSRowReader**, 141
- **restore**:backupFile:
  - **STSSparksee**, 165
- **run**: 26
  - **STSCSVWriter**, 40
  - **STSCSVReader**, 35
  - **getMaxLengthString**, 15
  - **getMean**, 15
  - **getMedian**, 15
  - **getMin**, 16
  - **getMinLengthString**, 16
  - **getMode**, 16
  - **getModeCount**, 16
  - **getNull**, 16
  - **getTotal**, 16
  - **getVariance**, 17
  - **STSSparksee**, 165
  - **STSCSVReader**, 35
  - **getCount**, 9
  - **getDataType**, 9
  - **getId**, 9
  - **getKind**, 9
  - **getName**, 9
  - **getSize**, 9
  - **getTypeId**, 9
  - **isSessionAttribute**, 9
- **STSAttributeList**, 10
  - **add:** 11
  - **count**, 11
  - **init**, 11
- **STSAttributeListIterator**, 11
  - **hasNext**, 13
  - **next**, 13
- **STSAttributeStatistics**, 13
  - **getAvgLengthString**, 15
  - **getDistinct**, 15
  - **getMax**, 15
- **runNPhases**: 22
  - **STSEdgeTypeLoader**, 71
  - **STSNodeTypeLoader**, 110
  - **runTwoPhases**: 22
  - **STSEdgeTypeLoader**, 71
  - **STSNodeTypeLoader**, 110
  - **STSTypeLoader**, 217
- **runSSparksee**: 26
  - **STSGraph**, 92
  - **STSSparksee**, 165
  - **count**, 18
  - **init**, 18
- **STSSparksee**, 165
  - **hasNext**, 20
  - **next**, 20
  - **STSSparksee**, 165
  - **runNPhases**: 22
  - **STSEdgeTypeLoader**, 71
  - **STSNodeTypeLoader**, 110
  - **runTwoPhases**: 22
  - **STSEdgeTypeLoader**, 71
  - **STSNodeTypeLoader**, 110
  - **STSTypeLoader**, 217
  - **getMaxLengthString**, 15
  - **getMean**, 15
  - **getMedian**, 15
  - **getMin**, 16
  - **getMinLengthString**, 16
  - **getMode**, 16
  - **getModeCount**, 16
  - **getNull**, 16
  - **getTotal**, 16
  - **getVariance**, 17
  - **STSCSVReader**, 38
  - **STSRowReader**, 141
  - **retrieve**:backupFile:
  - **STSSparksee**, 165
  - **run**: 26
  - **STSEdgeTypeLoader**, 71
  - **STSNodeTypeLoader**, 110
  - **STSTypeLoader**, 217
setFrequency:, 107
setGraph:, 107
setHeader:, 107
setRowWriter:, 108
setType:, 108

STSNodeTypeLoader, 108
initWithRowReader:graph:type:attrs:attrsPos:, 110
registerListener:, 110
run, 110
runNPhases:, 110
runTwoPhases, 111
setAttributePositions:, 111
setAttributes:, 111
setFrequency:, 111
setGraph:, 111
setLocale:, 111
setLogError:, 112
setLogOff, 112
setRowReader:, 112
setTimestampFormat:, 112
setType:, 112

STSObjects, 112
add:, 115
any, 115
clone, 115
cloneWithObjects:, 115
close, 115
combineDifference:objs2:, 115
combineIntersection:objs2:, 116
combineUnion:objs2:, 116
contains:, 116
count, 117
difference:, 117
equals:, 117
exists:, 117
intersection:, 117
iterator, 118
iteratorFromElement:, 118
iteratorFromIndex:, 118
remove:, 118
sample:samples:, 118
union:, 119

STSObjectsIterator, 119
close, 120
hasNext, 120
next, 120

STSOidList, 121
add:, 122
count, 122
init, 122
initWithNumInvalidOIDs:, 122
set:oid:, 123

STSOidListIterator, 123
hasNext, 124
next, 124

STSPlatform, 125
getStatistics:, 126

STSPlatformStatistics, 126
getAvailableMem, 127
getNumCPUs, 127
getRealTime, 127
getSystemTime, 127
getTotalMem, 127
generatesUserTime, 128

STSScriptParser, 143
generateSchemaScript:db:, 144
parse:execute:localeStr:, 145
setErrorLog:, 145
setOutputLog:, 145

STSSession, 146
close, 147
createObjects, 147
getGraph, 147

STSShortestPath, 147
addAllEdgeTypes:, 148
addEdgeType:dir:, 149
addNodeType:, 149
close, 149
excludeEdges:, 149
excludeNodes:, 150
run, 150
setMaximumHops:, 150

STSSinglePairShortestPath, 150
addAllEdgeTypes:, 152
addEdgeType:dir:, 152
addNodeType:, 152
close, 153
excludeEdges:, 153
excludeNodes:, 153
getCost, 153
getPathAsEdges, 153
generateNodes, 153
run, 154
setMaximumHops:, 154
STSSinglePairShortestPathBFS, 154
addAllEdgeTypes:, 156
addEdgeType:dir:, 156
addNodeType:, 156
checkOnlyExistence, 156
close, 156
excludeEdges:, 157
excludeNodes:, 157
getCost, 157
getPathAsEdges, 157
generateNodes, 157
initWithSession:src:dst:, 157
setMaximumHops:, 158
STSSinglePairShortestPathDijkstra, 158
addAllEdgeTypes:, 160
addEdgeType:dir:, 160
addNodeType:, 160
addWeightedEdgeType:dir:attr:, 160
close, 161
excludeEdges:, 161
excludeNodes:, 161
getCost, 161
getPathAsEdges, 161
generateNodes, 161
initWithSession:src:dst:, 162
setMaximumHops:, 162
setUnweightedEdgeCost:, 162
STSSparksee, 162
create:alias:, 164
initWithConfig:, 164
open:readOnly:, 164
restore:backupFile:, 165
STSSparkseeConfig, 165
generateCacheMaxSize, 169
generateCacheStatisticsEnabled, 169
generateCacheStatisticsFile, 169
generateCacheStatisticsSnapshotTime, 169
generateExtentPages, 169
generateExtentSize, 170
generateHighAvailabilityCoordinators, 170
generateHighAvailabilityEnabled, 170
generateHighAvailabilityIP, 170
generateHighAvailabilityMasterHistory, 170
generateHighAvailabilitySynchronization, 170
generateLicense, 170
generateLogFile, 171
generateLogLevel, 171
getPoolClusterSize, 171
getPoolFrameSize, 171
getPoolPersistentMaxSize, 171
getPoolPersistentMinSize, 171
getPoolTemporaryMaxSize, 171
getPoolTemporaryMinSize, 172
getRecoveryCacheMaxSize, 172
getRecoveryCheckpointTime, 172
generateRecoveryEnabled, 172
generateRecoveryLogFile, 172
generateRollbackEnabled, 172
init, 172
generateCacheMaxSize:, 172
generateCacheStatisticsEnabled:, 173
generateCacheStatisticsFile:, 173
generateCacheStatisticsSnapshotTime:, 173
generateExtentPages:, 173
generateExtentSize, 173
generateHighAvailabilityCoordinators:, 173
generateHighAvailabilityEnabled:, 174
generateHighAvailabilityIP:, 174
generateHighAvailabilityMasterHistory:, 174
generateHighAvailabilitySynchronization:, 174
generateLicense:, 174
generateLogFile:, 174
generateLogLevel:, 174
generatePoolClusterSize:, 175
generatePoolFrameSize:, 175
generatePoolPersistentMaxSize:, 175
generatePoolPersistentMinSize:, 175
generatePoolTemporaryMaxSize:, 175
generatePoolTemporaryMinSize:, 175
generateRecoveryCacheMaxSize:, 175
generateRecoveryCheckpointTime:, 176
generateRecoveryEnabled:, 176
generateRecoveryLogFile:, 176
generateRollbackEnabled:, 176
STSSparkseeProperties, 176
generate:def:, 178
generateBoolean:def:, 178
generateInteger:def:, 178
generateTimeUnit:def:, 178
load:, 179
STStringLengthList, 179
generate:add:, 180
generate:count, 180
generate:init, 181
STStringLengthListIterator, 181
hasNext, 182
next, 182
STSStrongConnectivity, 183
generate:AddAllEdgeTypes:, 185
generate:AddEdgeType:dir:, 185
generate:AddNodeType:, 185
close, 185
generate:ExcludeEdges:, 185
generate:ExcludeNodes:, 185
getConnectedComponents, 186
run, 186
s变态ializedAttribute:, 186
STSStrongConnectivityGabow, 186
addAllEdgeTypes:, 188
addEdgeType:dir:, 189
addNodeType:, 189
close, 189
excludeEdges:, 189
excludeNodes:, 189
getcOnnectedComponents, 189
initWithSession:, 189
setMaterializedAttribute;, 190
STSTextStream, 190
close, 192
initWithAppend:, 192
IsNull, 192
readString:, 192
writeString:, 192
STSTraversal, 192
addAllEdgeTypes:, 194
addEdgeType:dir:, 194
addNodeType:, 194
close, 194
excludeEdges:, 195
excludeNodes:, 195
getcOnnectedDepth, 195
hasNext, 195
next, 195
setMaximumHops:, 195
STSTraversalBFS, 196
addAllEdgeTypes:, 197
addEdgeType:dir:, 197
addNodeType:, 198
close, 198
excludeEdges:, 198
excludeNodes:, 198
getcOnnectedDepth, 198
hasNext, 198
initWithSession:node:, 198
next, 199
setMaximumHops:, 199
STSTraversalDFS, 199
addAllEdgeTypes:, 201
addEdgeType:dir:, 201
addNodeType:, 201
close, 201
excludeEdges:, 201
excludeNodes:, 201
getcOnnectedDepth, 201
hasNext, 202
next, 202
setMaximumHops:, 202
STSType, 203
geteAreNeighborsIndexed, 204
getId, 204
getIsDirected, 204
getIsRestricted, 204
getName, 205
getNumObjects, 205
getObjectType, 205
getRestrictedFrom, 205
getRestrictedTo, 205
STSTypeExporter, 205
registerListener:, 207
run, 207
setAttributes:, 207
setFrequency:, 207
setGraph:, 207
setHeader:, 208
setRowWriter:, 208
setType:, 208
STSTypeExporterEvent, 208
getcOunt, 209
getTotal, 209
getTypeld, 209
isLast, 210
STSTypeExporterListener, 210
notifyEvent:, 211
STSTypeList, 211
add:, 212
count, 212
init, 213
STSTypeListIterator, 213
hasNext, 214
next, 214
STSTypeLoader, 215
registerListener:, 216
run, 216
runNPhases:, 217
runTwoPhases, 217
setAttributesPositions:, 217
setAttributes:, 217
setFrequency:, 217
setGraph:, 218
setLocale:, 218
setLogError:, 218
setLogOff, 218
setRowReader:, 218
setTimestampFormat:, 218
setType:, 219
STSTypeLoaderEvent, 219
getcOunt, 220
getPartition, 220
getPhase, 220
getTotalPartitionSteps, 221
getTotalPartitions, 220
getTotalPhases, 221
getTypeld, 221
isLast, 221
STSTypeLoaderListener, 221
notifyEvent:, 222
STSTypeValue, 223
compaare:, 225
equals:, 225
getBoolean, 226
getDataType, 226
getDouble, 226
getInteger, 226
getLong, 226
getOid, 226
getString, 227
gETimestamp, 227
gETimestampAsNSDate, 227
init, 227
initWithValue:, 227
isNull, 227
set:, 227
setBoolean:, 228
setBooleanVoid:, 228
setDouble:, 228
setDoubleVoid:, 228
setInteger:, 228
setIntegerVoid:, 229
setLong:, 229
setLongVoid:, 229
setNull, 229
setOid:, 229
setOidVoid:, 229
setString:, 229
setStringVoid:, 229
setTimestamp:, 229
setTimestampVoid:, 229
setTimestampVoidWithYear:month:day:hour:minutes:seconds:millisecs:, 229
setTimestampWithNSDate:, 229
setTimestampWithYear:month:day:hour:minutes:seconds:milliseconds:, 229
setVoid:, 230

STSValueList, 231
add:, 231
count, 231
gET, 231
init, 231

STSTvalueListIterator, 233
hasNext, 233
next, 233

STSTvalues, 235
close, 235
count, 235

STSTvaluesIterator, 235
hasNext, 235
next, 235

STSTweakConnectivity, 238
addAllEdgeTypes, 240
addEdgeType:, 240
addNodeType:, 240
close, 240
excludeEdges:, 240
excludeNodes:, 240
getConnectedComponents, 240
run, 240

setMaterializedAttribute:, 241

STSWeakConnectivityDFS, 242
addAllEdgeTypes, 244
addEdgeType:, 244
addNodeType:, 244
close, 244
excludeEdges:, 244
excludeNodes:, 245
gETConnectedComponents, 245
initWithSession:, 245
setMaterializedAttribute:, 245

SWIGTYPE_p_wchar_t, 246

sample:samples:
STSTObjects, 118

selectWithAttrValue:cond:value:
STSTGraph, 92

selectWithAttrValueRestriction:cond:value:restriction:
STSTGraph, 93

selectWithAttrValues:cond:lower:higher:
STSTGraph, 93

selectWithAttrValuesRestriction:cond:lower:higher:restriction:
STSTGraph, 93

selectWithType:
STSTGraph, 94

set:
STSTGraph, 227

set:oid:
STSTOldList, 123

setAsDirected:
STSTEdgeExport, 63

setAttribute:attr:value:
STSTGraph, 94

setAttributeDefaultValue:value:
STSTGraph, 94

setAttributePositions:
STSTEdgeTypeLoader, 71
STSTNodeTypeLoader, 111
STSTypeLoader, 217

setAttributeText:attr:tstream:
STSTGraph, 94

setAttributes:
STSTEdgeTypeExporter, 66
STSTEdgeTypeLoader, 71
STSTNodeTypeExporter, 107
STSTNodeTypeLoader, 111
STSTypeExporter, 207
STSTypeLoader, 217

setAutoQuotes:
STSTCSVWriter, 42

setBoolean:
STSTValue, 228

setBooleanVoid:
STSTValue, 228

setCacheMaxSize:
STSTDatabase, 46
STSTSparkseeConfig, 172

setCacheStatisticsEnabled:
setCacheStatisticsFile:
STSSparkseeConfig, 173
setCacheStatisticsSnapshotTime:
STSSparkseeConfig, 173

setColorRGB:
STSEdgeExport, 63
STSNodeExport, 103
setColorRed:green:blue:alpha:
STSEdgeExport, 63
STSNodeExport, 103

setDouble:
STSVvalue, 228
setDoubleVoid:
STSVvalue, 228

setErrorLog:
STSScriptParser, 145

setExtentPages:
STSSparkseeConfig, 173
setExtentSize:
STSSparkseeConfig, 173

setFit:
STSSparkseeConfig, 173

setFontSize:
STSEdgeExport, 63
STSNodeExport, 104

setForcedQuotes:
STSCSVWriter, 42

setFrequency:
STSEdgeTypeExporter, 66
STSEdgeTypeLoader, 71
STSNodeTypeExporter, 107
STSNodeTypeLoader, 111
STSTypeExporter, 207
STSTypeLoader, 217

setGraph:
STSEdgeTypeExporter, 67
STSEdgeTypeLoader, 71
STSNodeTypeExporter, 107
STSNodeTypeLoader, 111
STSTypeExporter, 207
STSTypeLoader, 218

setHeadAttribute:
STSEdgeTypeExporter, 67
STSEdgeTypeLoader, 72

setHeadPosition:
STSEdgeTypeExporter, 67
STSEdgeTypeLoader, 72

setHeader:
STSEdgeTypeExporter, 67
STSNodeTypeExporter, 107
STSTypeExporter, 208

setHeight:
STSNodeExport, 104

setHighAvailabilityCoordinators:
STSSparkseeConfig, 173

setHighAvailabilityEnabled:
STSSparkseeConfig, 174
setHighAvailabilityIP:
STSSparkseeConfig, 174
setHighAvailabilityMasterHistory:
STSSparkseeConfig, 174
setHighAvailabilitySynchronization:
STSSparkseeConfig, 174

setInteger:
STSVvalue, 228
setIntegerVoid:
STSVvalue, 229

setLabel:
STSEdgeExport, 63
STSGraphExport, 96
STSNodeExport, 104

setLabelColorRGB:
STSEdgeExport, 63
STSNodeExport, 104

setLabelColorRed:green:blue:alpha:
STSEdgeExport, 63
STSNodeExport, 104

setLicense:
STSSparkseeConfig, 174

setLocale:
STSCSVReader, 39
STSCSVWriter, 42
STSEdgeTypeLoader, 72
STSNodeTypeLoader, 111
STSTypeLoader, 218

setLogFile:
STSSparkseeConfig, 174

setLogError:
STSEdgeTypeLoader, 72
STSNodeTypeLoader, 112
STSTypeLoader, 218

setLogFile:
STSSparkseeConfig, 174

setLogLevel:
STSSparkseeConfig, 174

setLogOff:
STSEdgeTypeLoader, 72
STSNodeTypeLoader, 112
STSTypeLoader, 218

setLong:
STSVvalue, 229
setLongVoid:
STSVvalue, 229

setLookAhead:
STSCommunitiesSCD, 23

setMaterializedAttribute:
STSCommunitiesSCD, 23
STSCommunity, 31
STSDisjointCommunityDetection, 57
STSStrongConnectivity, 186
STSStrongConnectivityGabow, 190
STSSWeakConnectivity, 241
STSSWeakConnectivityDFS, 245

setMaximumHops:
STSShortestPath, 150

Generated on Tue Jun 2 2015 15:49:36 for Sparksee by Doxygen
setStartLine:
  STSCSVReader, 40
setStream:handler:
  STSScriptParser, 145
setString:
  STSValue, 230
setStringValue:
  STSValue, 230
setTailAttribute:
  STSEdgeTypeExporter, 67
  STSEdgeTypeLoader, 73
setTailPosition:
  STSEdgeTypeExporter, 67
  STSEdgeTypeLoader, 73
setTimestamp:
  STSValue, 230
setTimestampFormat:
  STSEdgeTypeLoader, 73
  STSNodeTypeLoader, 112
  STSTypeLoader, 218
setTimestampVoid:
  STSValue, 230
setTimestampVoidWithYear:month:day:hour:minutes:
  seconds:milliseconds:
  STSValue, 231
setTimestampWithNSDate:
  STSValue, 230
setTimestampWithYear:month:day:hour:minutes:
  seconds:milliseconds:
  STSValue, 231
setUnweightedEdgeCost:
  STSSinglePairShortestPathDijkstra, 162
setVoid:
  STSValue, 231
setWidth:
  STSEdgeExport, 64
  STSNodeExport, 105
start:
  STSScriptStream, 132
tails:
  STSGraph, 94
tailsAndHeads:tails:heads:
  STSGraph, 95
union:
  STSObjects, 119
write:
  STSCSVWriter, 43
  STSRowWriter, 143
writeString:
STSTextStream, 192