

Table of Contents

Optics Interfaces.....	1
Currents production.....	1
Beta star processing.....	1
Currents processing.....	1
Magnet strength processing.....	2
Optics Generation.....	2
Elastic Optics.....	2
Inelastic optics.....	3

Optics Interfaces

Service name: *OpticsManager*

Currents production

Beta star processing

- **int saveBetaStar(long startTime, long endTime, double betaStar)**
Description: Saves single beta star value into the database. The method returns version number of saved data (in case of multiple similar inserts the old one is not removed, but new version of the same data is created instead).
Parameters:
 - *startTime* - validity interval start time. Nonnegative long value.
 - *endTime* - validity interval end time. Nonnegative long value.
 - *betaStar* - beta star value
- **double loadBetaStar(long timestamp)**
Description: Loads latest version of single beta star value for given timestamp from the database.
Parameters:
 - *timestamp* - timestamp, nonnegative long value.
- **double loadBetaStar(long timestamp, int version)**
Description: Loads specific version of single beta star value for given timestamp from the database.
Parameters:
 - *timestamp* - timestamp. Nonnegative long value.
 - *version* - version number. Nonnegative integer value. If 0, the latest version is loaded.

Currents processing

- **int saveCurrents(long startTime, long endTime, map<string, double> currents)**
Description: Saves currents data into the database. The method returns version number of saved data (in case of multiple similar inserts the old one is not removed, but new version of the same data is created instead).
Parameters:
 - *startTime* - validity interval start time. Nonnegative long value.
 - *endTime* - validity interval end time. Nonnegative long value.
 - *currents* - label - value currents map. Keys of the map (strings) cannot be null or empty strings.
- **map<string, double> loadCurrents(long timestamp)**
Description: Loads latest version of currents from the database by given timestamp and returns them as label-value map.
Parameters:
 - *timestamp* - timestamp value. Nonnegative long value.
- **map<string, double> loadCurrents(long timestamp, int version)**
Description: Loads specific version of currents from the database by given timestamp and returns them as label-value map.
Parameters:
 - *timestamp* - timestamp value. Nonnegative long value.
 - *version* - version number. Nonnegative integer value. If 0, the latest version is loaded.
- **map<string, double> loadAverageCurrents(long startTime, long endTime)**
Description: Loads latest version of currents from the database and returns them as label-value map. The values are averaged for given time interval.

Parameters:

- *startTime* - time interval start time. Nonnegative long value.
- *endTime* - time interval end time. Nonnegative long value.

- **map<string, double>** *loadAverageCurrents(long startTime, long endTime, int version)*

Description: Loads latest version of currents from the database and returns them as label-value map. The values are averaged for given time interval.

Parameters:

- *startTime* - time interval start time. Nonnegative long value.
- *endTime* - time interval end time. Nonnegative long value.
- *version* - version number. Nonnegative integer value. If 0, the latest version is loaded.

Magnet strength processing

- **int** *saveMagnetStrength(long startTime, long endTime, map<string, double> magnetStrength)*

Description: Saves magnet strength value map into the database. The method returns version number of saved data (in case of multiple similar inserts the old one is not removed, but new version of the same data is created instead).

Parameters:

- *startTime* - validity interval start time. Nonnegative long value.
- *endTime* - validity interval end time. Nonnegative long value.
- *magnetStrength* - magnet label - measurement value map. Keys of the map (strings) cannot be null or empty strings. Labels which can be used are specified here.

- **map<string, double>** *loadMagnetStrength(long timestamp)*

Description: Loads latest version of magnet strength from the database by given timestamp and returns them as label-value map.

Parameters:

- *timestamp* - timestamp. Nonnegative long value.

- **map<string, double>** *loadMagnetStrength(long timestamp, int version)*

Description: Loads specific version of magnet strength from the database by given timestamp and returns them as label-value map.

Parameters:

- *timestamp* - timestamp. Nonnegative long value.
- *version* - version number. Nonnegative integer value. If 0, the latest version is loaded.

Optics Generation

Elastic Optics

- **int** *saveElasticOptics(long startTime, map<string, map<string, double>> opticsMatrix)*

Description: Saves elastic optics matrix into the database. Method returns version number of saved data (in case of multiple similar inserts the old one is not removed, but new version of the same data is created instead).

Parameters:

- *startTime* - validity interval start (and the end of validity interval of previously inserted data). Nonnegative long value.
- *opticsMatrix* - elastic optics matrix map. The keys are string labels describing third dimension of matrix (i. e. MBCS2.1L5), the values are also maps of optics in two-dimensional matrix bound to string labels. Keys in both maps (strings) cannot be null or empty strings.

- **map<string, map<string, double>>** *loadElasticOptics*(**long** timestamp)
Description: Loads latest version of elastic optics matrix from the database by given timestamp.
Parameters:
– *timestamp* - timestamp. Nonnegative long value.
- **map<string, map<string, double>>** *loadElasticOptics*(**long** timestamp, **int** version)
Description: Loads specific version of elastic optics matrix from the database by given timestamp.
Parameters:
– *timestamp* - timestamp. Nonnegative long value.
– *version* - version number. Nonnegative integer value. If 0, the latest version is loaded.

Inelastic optics

- **int** *saveInelasticOpticFile*(**long** startTime, **long** endTime, **FileType** type, **string** pathToFile)
Description: Saves inelastic optic file into the database. Method returns version number of saved file (in case of multiple similar inserts the old one is not removed, but new version of the same file is created instead).
Parameters:
– *startTime* - a validity interval start time (UTC time value). Nonnegative long value.
– *endTime* - a validity interval end time (UTC time value). Nonnegative long value.
– *type* - an enumeration type distinguishing reconstruction and simulation files
– *pathToFile* - path to file, which will be saved into the database. Not null value.
- **int** *loadInelasticOpticFile*(**long** timestamp, **FileType** type, **string** pathToFile)
Description: Loads latest version of inelastic optic file from the database. The file is saved in given path on local file system. User should specify filename as well, i.e. pathToFile=./example.file means that file will be saved in current directory with example.file name.
Parameters:
– *timestamp* - a UTC timestamp. Nonnegative long value.
– *type* - an enumeration type distinguishing reconstruction and simulation files
– *pathToFile* - path to file, where loaded data will be saved. All directories on the path should exist. If file with the same path exist on local file system, it will be overwritten. Not null value.
- **int** *loadInelasticOpticFile*(**long** timestamp, **FileType** type, **string** pathToFile, **int** version)
Description: Loads specific version of inelastic optic file from the database. The file is saved in given path on local file system. User should specify filename as well, i.e. pathToFile=./example.file means that file will be saved in current directory with example.file name.
Parameters:
– *timestamp* - a UTC timestamp. Nonnegative long value.
– *type* - an enumeration type distinguishing reconstruction and simulation files
– *pathToFile* - path to file, where loaded data will be saved. All directories on the path should exist. If file with the same path exist on local file system, it will be overwritten. Not null value.
– *version* - a version number. Nonnegative integer value. If 0, the latest version is loaded.
- **vector<OpticFileMetadata>** *listOpticFiles*(**long** startTime, **long** endTime, **FileType** type)
Description: List files and their version for given type and period of time.
Parameters:

OpticsInterfaces < TOTEM < TWiki

- *startTime* - a start of period of time for which data will be returned. Nonnegative long value.
- *endTime* - a end of period of time for which data will be returned. Nonnegative long value.
- *type* - files' type

- **enumeration** *FileType*

Description: An enumeration type distinguishing reconstruction and simulation files

Values:

- *RECO* - a reconstruction file type
 - *SIM* - a simulation file type
-

This topic: TOTEM > OpticsInterfaces

Topic revision: r7 - 2012-11-22 - unknown



Copyright &© 2008-2022 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

or Ideas, requests, problems regarding TWiki? use [Discourse](#) or [Send feedback](#)