NAME

ts_config - trace-shell configuration file format

DESCRIPTION

This documents version 2.0.0 of the **trace-shell** configuration file format.

This is documentation for the file which the telescope configuration which will be raytraced by **trace-shell**. **trace-shell** raytraces a single Wolter type I X-ray telescope shell with various apertures and baffles.

File Format

Config::Wild is used to parse the file; see its documentation for more information on the format. The file must have a suffix of `.cnf`.

Boolean keywords take `yes` or `no` values. The keywords are specified below in the form which **trace-shell** will query for them. For example "mirror_(geo)(shell)_dfm2_db" indicates that the keyword name will be formed using the `geo` and `shell` variables. The configuration file may specify either exact matches or may use the form above; see the documentation for **Config::Wild** for more information.

Input Variables

trace-shell will set the following keywords before parsing the configuration file; they may be used within the configuration file where necessary.

`shell` *integer*

the shell being raytraced

`geo` *string*

either `p` or `h` depending upon whether it is looking for a keyword for a paraboloid or hyperboloid.

Required keywords

The following keywords are required.

`micro_scatter` *boolean*

Indicates whether scattering due to surface micro-roughness should be simulated. It requires that the keywords `hdos_xform`, and `micro_scat_db` be set.

`micro_scat_$(geo)$(shell)_db` *string*

A scattering database specific to the optic.

`scatter_shield` *boolean*

Indicates whether a post-HRMA scatter shield should be simulated. Use this *only* if the thermal post-collimator is *not* installed, scattering is turned on, and you want to avoid the few spurious rays which may scatter across the optical axis and "through" the other side of the optic. If this is used, the `mirror_geo_db` file must contain the columns `z_a` and `rho_a`, which are the position of the aft edge hyperboloid and its radius.

`cap` *boolean*

Indicates whether the Central Aperture Plate should be simulated. The `cap_spec` keyword must be set.

`ghostbaffle` *boolean*

Indicates whether the mirror shell 6 ghost baffle should be simulated. The `ghostbaffle_spec` keyword must be set.

`precoll` *boolean*

Indicates whether the thermal pre-collimator should be simulated. The `precoll_spec` keyword should be set.

`postcoll` *boolean*

Indicates whether the thermal post-collimator should be simulated. The `postcoll_spec` keyword be set.

`entrance_ap_db` *filename*

The name of an entrance aperture database compatible with the **raygen** configuration script specified in the `raygen_default_cfg` and `raygen_focus_cfg` keywords.

`mirror_geo_db` *filename*

An **RDB** file containing mirror position and tilt information. The **RDB** file must have at least these columns: `mirror`, `x0`, `y0`, `z0`, `p`, `k`, `rho0`, `theta0`, `az_mis`, `el_mis`, `l`. `mirror` is the name of the optic, e.g., "p1". The rest are OSAC parameters. The file *must* have a suffix of `.rdb`.

`reflect_$(geo)$(shell)_db` *filename*

A **reflect** compatible surface description file, which will be passed to **reflect** via the `rfldata_filename` parameter.

`mirror_$(geo)$(shell)_dfm_db` *filename*

A **SAOdrat** compatible mirror deformation file. As indicated by the keyword name, the actual keyword requested will have the geometry (`p` or `h`) and shell number embedded in it.

Optional Keywords

The following keywords are optional. Some are ancillary to the above keywords, and may be required by the settings of the above keywords. Others are stand-alone.

`mirror_$(geo)$(shell)_dfm_scale` *float*

The scale factor for the primary mirror deformation. It defaults to `1.0`.

`mirror_$(geo)$(shell)_dfm2_db` *filename*

The secondary mirror deformation file. This must be a Fourier - Legendre coefficients file.

`mirror_$(geo)$(shell)_dfm2_scale` *float*

The scale factor for the secondary mirror deformation. It defaults to `1.0`.

`micro_scat_algorithm` *string*

If present, the new scattering code which reads FITS formatted scattering table is used. If set to `old`, the new code emulates the old scattering algorithm (LVS's). If present and set to anything else, the new code uses the new algorithm (P. Zhao's). If not present, the old scattering code which uses native formatted binary files is used.

raygen_default_cfg *filename or Lua module name*

a **raygen** compatible configuration specification script. This is only used if the **trace-shell** parameter `src` is set to `default`.

raygen_focus_cfg *filename or Lua module name*

a **raygen** compatible source specification script to be used for focus runs. This is only

used if the **trace-shell** parameter `src` is set to `default` and the `focus` parameter is true.

scatter_shield_spec *filename or Lua module name*

precoll_spec *filename or Lua module name*

cap_spec *filename or Lua module name*

postcoll_spec *filename or Lua module name*

ghostbaffle_spec *filename or Lua module name*

aperture compatible scripts to model the various apertures in the system. The variables

```
shell
assembly_name
config_db
```

will be set in a Lua script passed to **aperture** via its `cfgpars` parameter.

micro_scat_db *filename*

The `RDB` table to be passed to the **scatter** program via its `scatter_db` parameter.

hdos_xform_db *filename*

The `RDB` table to be passed to the **scatter** program via its `hdos_xform_db` parameter.

SEE ALSO

Config::Wild, *trace-shell*

AUTHOR

Diab Jerius <djerius@cfa.harvard.edu>