%xuprogpath

Overview

The `%xuprogpath` macro determines the full path of the currently executing SAS program and creates global variables containing task-related information based on Atorus's standard programming environment structure. It parses the program path to extract information about the client, compound, protocol, task, and other organizational elements.

Version Information

- Version: 1.0
- Last Updated: 19FEB2021
- Author: Atorus Research

Dependencies

- SAS Version: SAS 9.4 V9
- Required Global Variables:
  - `_sasprogramfile`: Set by SAS EG to contain the full program path
  - `__root`: Root directory path (set in autoexec.sas)
  - `__sponsor_level`: Flag indicating sponsor-level directory structure (Y/N; set in autoexec.sas)
  - `__prod_qc_separation`: Flag indicating production/QC separation (Y/N; set in autoexec.sas)
  - `II`: Directory separator character (set in autoexec.sas)
- No macro dependencies

Parameters

None - The macro uses global variables and system information to perform its functions.

Return Values/Output

Creates the following global variables:

- **Path-related**:
  - `__program_full_path`: Full path to the executing program
  - `__p_name`: Program name without extension
  - `__p_path`: Program directory path
  - `__log_path`: Path for log files
  - `__lst_path`: Path for list files
- **Organization-related**:
  - `__clnt`: Client name (if sponsor_level=Y)
  - `__comp`: Compound name
  - `__prot`: Protocol name
  - `__subfolders`: Intermediate folder structure

- `__task`: Task name
- `__level`: Development level (development/final)
- `__side`: Production side (prod/val)
- `__type`: Program type (sdtm/adam/tlf)

Processing Details

1. Environment check:
   - Verifies existence of `_sasprogramfile` variable
   - Validates program path is not empty
2. Path parsing:
   - Extracts program path relative to root directory
   - Determines organizational structure based on `__sponsor_level`
   - Identifies development level (final/development)
   - Extracts subfolder structure between protocol and task
3. Variable creation:
   - Sets path-related variables
   - Determines organizational variables
   - Creates log and list file paths
4. Logging:
   - Outputs all created variables to log
   - Groups information by category (program, task, outputs)

Examples

```
/* Basic usage - typically called by [`setup.sas`](/man/study_specific/setup.md)
or %xumprint */
%xuprogpath;

/* Example output in log:
********************************************************************************
********************
=== Program being executed ===
Program executed by: jsmith
Program full path  :
C:/Projects/Client/Compound/Protocol/Task/final/sdtm/program/dm.sas
Program name       : dm
Program path       : C:/Projects/Client/Compound/Protocol/Task/final/sdtm/program/
********************************************************************************
********************
=== Task Information ===
Client    : Client
Compound  : Compound
Protocol  : Protocol
```

```
Subfolders:
Task      : Task
Level     : final
Side      : prod
Type      : sdtm
********************************************************************************
********************
=== Logs and lst outputs path ===
Log path: C:/Projects/Client/Compound/Protocol/Task/final/sdtm/log/
Lst path: C:/Projects/Client/Compound/Protocol/Task/final/sdtm/lst/
********************************************************************************
********************
*/
```

Common Issues and Solutions

1. **Missing Program Path**
   - Issue: `_sasprogramfile` is empty or not set
   - Solution: Ensure program is run through SAS EG or proper environment setup
2. **Incorrect Directory Structure**
   - Issue: Unable to parse organizational elements
   - Solution: Verify folder structure follows Atorus standards
3. **Missing Global Variables**
   - Issue: Required global variables not set
   - Solution: Ensure autoexec.sas properly initializes environment

Notes and Limitations

1. Designed specifically for Atorus's standard programming environment structure.
2. Requires specific directory hierarchy:
   - Optional sponsor level: Client/Compound/Protocol
   - Standard level: Compound/Protocol
3. Assumes standard folder types (development/final) and program types (sdtm/adam/tlf).
4. Should be called automatically by `%xumprint` or `setup.sas`.

See Also

- `%xumprint`: Calls `%xuprogpath` for program information
- `setup.sas`: Sets up the study global variables and libnames
- Other setup and initialization macros

Change Log

Version 1.0 (19FEB2021)

- Initial release

- Basic path parsing functionality
- Support for sponsor-level directory structures
- Automatic log/lst path determination