

TR-369 / USP & TR-069 Protocol Stack

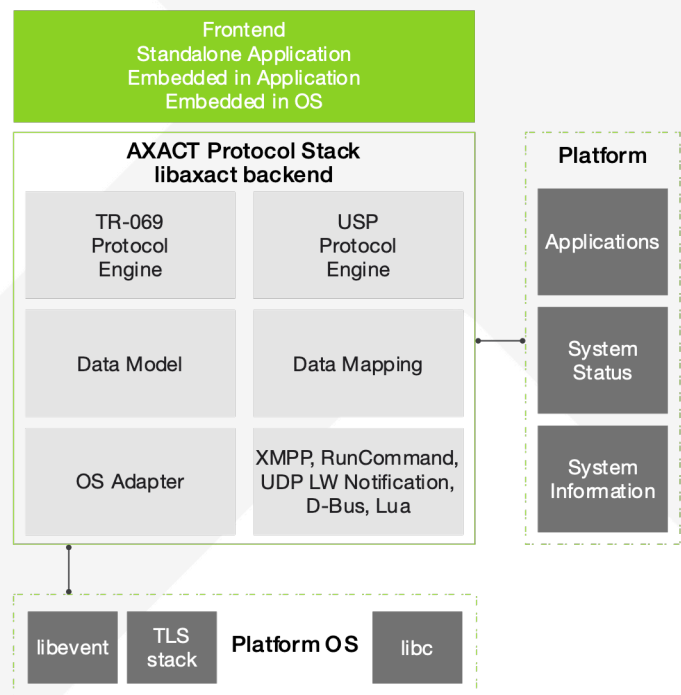
Axios AXACT provides a fully Broadband Forum (“BBF”) standards compliant TR-069 client and TR-369 / USP Agent protocol implementation. By means of a highly customizable data model, service providers can now configure, provision and monitor the underlying device features and functions.

Key Features of AXACT

- Designed to be powerful and extensible yet simple to use
- Single process event driven engine
- Very lightweight at around 150-400 kB binary size
- Always supplied in C++11 compatible C source code
- Extensive documentation, example apps and testing support
- Unified data model and mapping implementation
- Concurrent TR-069 and TR-369 protocol support

AXACT – Embedded Connectivity Features & Functionalities

- Works on every POSIX1.b compliant system
- May be integrated into existing applications, OS processes or tasks
- Modular design for easy customization
- Supports any TR-106 compliant model via automatic code generation
- IPv6 and dual stack operation supported
- TLS 1.2/1.3 capable with extensive certificate handling using the out-of-the-box
- Libre-, OpenSSL, and mbedTLS support or custom implementation
- UTF-8 support
- Embeddable Lua interpreter and bindings for scripting functionality
- Relies on libevent 2.x for communication, timers and buffer management





Embedded Connectivity | Device IoT-zation

User Services Platform (USP) Management Functions

All functions blocks are abstracted from the underlying operating system and utilize the libevent library which allows for easy porting to different target platforms. The backend functionality is activated and driven by a frontend. A variety of different frontend examples are included to base own implementations on.

The AXACT backend is structured into the main functional blocks:

- Protocol Engines (USP and/or TR-069)
- Data Model
- Data Mapper
- OS Adapter
- Optional functional modules for additional features

USP Protocol Engine

The USP Protocol Engine contains a full implementation of the USP standard. It comprises of all the requisite MTPs, record and message parsing and endpoint handling as well as processing of messages internally with connectivity to the notification system, data model, data mapper and frontend.

Data Model

The Data Model provides on one side a TR-106 compliant means of representing the internal device data to the ACS or Controller and on the other side towards the device it encapsulates all USP and TR-069 specific data and attribute handling. It ensures compliance of the values according to the datamodel specification (type, enumeration values, ranges or length, write-ability and read-ability, references), consistent storage and retrieval of metadata and attribute handling.

Function Modules

The functional modules provide extensions of core functionality and can be independently enabled or disabled or depending on the device's needs. In addition to the included functionality mentioned below, optional extension modules can be used to add further functionality to AXACT.

TR-069 Protocol Engine

The TR-069 Protocol Engine contains the process execution logic and provides complete TR-069 protocol handling. All internal and external events are processed within the event and session control parts of the TR-069 Protocol Engine. The execution of RPCs from the ACS is handled by the RPC subsystem which allows for easy customization of the supported RPC methods as well as the addition of vendor specific RPCs.

Data Mapper

The Data Mapper provides the connection between the Data Model, the system configuration and device state by calling pre-defined callbacks in order to fetch or set data as required by interaction with the USP Controller or TR-069 ACS. This is the part where typically the most customization is needed. Automatically generated stub functions for integration purposes are provided which can be implemented to connect to the data of a specific device.