FNT

// when transparency matters.

- Web service interfaces
- SOAP and REST protocol versions
- Process-based system integration
- Uni- and bi-directional interfaces
- Simple and complex data types
- Message queue

Command

// FNT Command API

The Predefined Interface for Easy Integration of External Software Systems and Applications

Modern system landscapes typically include hundreds of applications and software systems, often installed as multiple instances at multiple locations. A number of factors, such as the function-specific nature of individual software systems and their limited integration capabilities, can create internal silos that render valuable information inaccessible. When important data is isolated in this way, cross-departmental processes become increasingly inefficient. Full integration of the various systems and applications deployed within an organization is therefore an important strategic goal that increases the availability of data and enables seamless merging of cross-departmental processes.

The FNT Command API offers a huge range of possibilities for connecting and exchanging data between FNT Command and external systems. The extensive library of over 1,600 function calls can be used to integrate virtually every software system. With its release-independent web service interfaces, the integration technology delivers high-quality data as well as optimum stability when connecting with external applications. Extensive standardization means that interface failure rates and maintenance costs are significantly reduced, i.e., users can integrate systems across their organization with greater efficiency and reliability.

High-Performance Integration Technology

The FNT Business Gateway abstraction layer controls and coordinates all data access and query operations for FNT command. All changes to the data model are automatically made available to the API and are immediately valid for all subsequent function calls. In other words, the integration technology provides an additional layer of protection for the FNT Command database by supporting reliable, release-independent operation of interfaces as well as affordable customization.

Built-in plausibility checks use integrated business logic to ensure optimum data quality throughout the exchange process as well as correct allocation to the appropriate CIs.

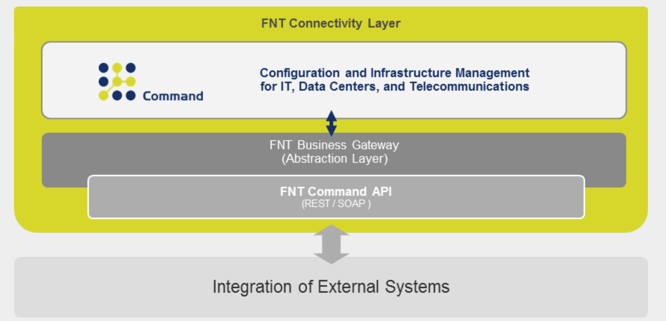

FNT Connectivity Layer
Command
Configuration and Infrastructure Management for IT, Data Centers, and Telecommunications
FNT Business Gateway (Abstraction Layer)
FNT Command API (REST / SOAP)
Integration of External Systems

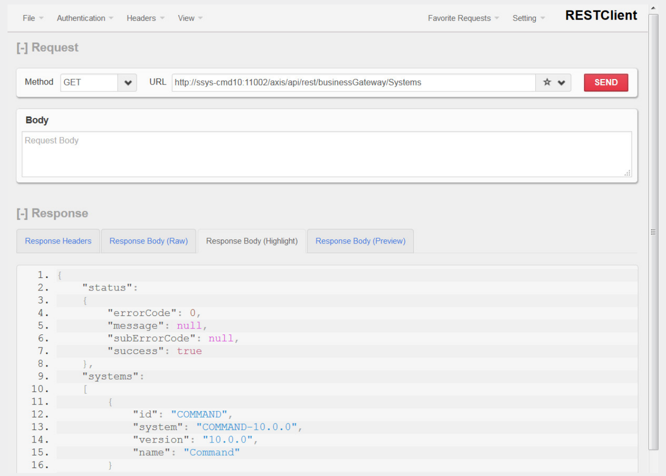
Fig. 1: Schematic view of interface architecture

Support for Diverse Integration Scenarios

All function calls available for application programming with FNT Command are documented in a central library with all methods, parameters, and return values in an HTML overview. Universal availability and usability of calls enables both one-off and mass integration of data. Bi-directional integration is event-based and available in batch mode to allow diverse integration scenarios, e.g., linking to workflow engines, use of orchestration tools, connection to dashboard systems, and integration of data from complex distributed systems.

Web services

Data can be exchanged between systems via web services using the SOAP protocol or the REST approach. This partial access to the data in FNT Command allows a wide range of function calls without the need to grant direct access rights to the user application.



Method GET URL http://ssys-cmd10:11002/axis/api/rest/businessGateway/Systems

```
{
    "status":
    {
        "errorCode": 0,
        "message": null,
        "subErrorCode": null,
        "success": true
    },
    "systems":
    [
        {
            "id": "COMMAND",
            "system": "COMMAND-10.0.0",
            "version": "10.0.0",
            "name": "Command"
        }
```

Fig. 2: Function call for a REST web service from FNT Command API

The number of executable methods can be set and defined via the web service interfaces. SOAP-based web service interfaces are defined using WSDL (Web Service Description Language), which allows platform-, language-, and protocol-independent exchange of XML-based messages. This enables easy integration of external applications for search and data queries as well as write access to FNT Command. Limiting the exchange to data only makes it easier to create interfaces as there is no need to create or define GUIs. Instead, the functionality available in the FNT Command GUI can be accessed and used directly via a web service interface.

Alternatively, it is possible to use the predefined function calls for REST-based web service interfaces. This approach is most commonly used when integrating third-party systems using the RAD methodology (rapid application development), which offers rapid results with minimal programming workload.

Automatic Documentation

The methods, parameters, and return values used for web services are automatically documented and made available to the administrator as a report. In the event of a fault or error, the administrator can view the current, complete documentation at any time in order to identify and resolve the issue with minimum delay.

System Requirements

The FNT Command API requires the FNT Command C base module plus all additional modules that are to be accessed via the interface. We also recommend that administrators attend a training course on efficient use of the FNT Command API.