Robust System Updates

- Non-Interfering Device Integration

New types of 4-20 mA/HART, FOUNDATION fieldbus, and PROFIBUS devices become available every month and new device versions are released every week. Installing and upgrading software always have an element of risk, especially on a central computer like a server. This means challenges for the system administrator. Fortunately, the Electronic Device Description Language (EDDL) technology was designed to keep the system up to date with new types and versions of devices without having to install software. This keeps the system very much more robust.

Life-Cycle Challenge

Control systems remain operational for 15 years or more. During this period of time new device types and new device versions will arrive in the plant.

Software Risk Management

There is always a risk new software, especially device driver programs, will conflict with existing software on the system. For this reason DCS manufacturers do not permit third-party software, device drivers included, to be installed on the system servers and operator consoles. For the same reason, plants have strict policies for software installation and upgrade on systems. This is especially so for device drivers because they are not stand-alone applications, they have complex component-container interactions closely coupled with other programs on the computer.

Driver Program Problem

Some device management software use device driver programs similar to the Microsoft Windows printer driver concept. When new device types or new device versions arrive in the plant, a device driver for the device (or whole device family) has to be installed on each maintenance station to support it. Like all Windows programs, each device driver consists of executable components such as EXE-files and shared DLL-files. Windows registry changes are also made at installation.

“DLL-hell”

Windows programs consist of multiple components installed as Dynamic Link Library (DLL) files. These files are shared among programs. If an upgrade or new program, updates a shared DLL component, other programs may not be compatible with this new version and may therefore fail. Moreover, when an unwanted program is uninstalled, it may in some cases remove DLL components used by other programs causing them to fail. This conflict is colloquially referred to as "*DLL-hell*".

Because a plant has many dozens of device types, a large number of device drivers have to be installed, increasing the risk of DLL-hell conflicts[1] arising. Restoring the system afterwards can be very time consuming. There are mechanisms in Windows to minimize this risk, but over time, as driver install and uninstall is unsuccessful or is aborted, problems may still occur, particularly considering many device types and frequent new versions requiring driver updates.

If a new driver program, for a new device type or version, replaces or removes a shared DLL component, a vital DCS function or another device driver may fail. This is the reason why third-

[1] It is impractical for the DCS manufacturer to test installation, upgrade, and uninstalling of every device driver from every device manufacturer in different order.

party software is not allowed on a DCS, and why device management software based on device drivers is not permitted on the DCS server or operator console, but requires a dedicated maintenance station.

EDDL: The HTML of Process Control

HTML

Web pages use Hyper Text Markup Language (HTML) which is a text file that describes the user interface as plain text. The beautiful user interface graphics is then rendered by the web browser. That is, an HTML file is a document, not a software. The HTML pages are not installed by web browser (they are copied to a cache). Therefore web pages do not overwrite software components or Windows registry on the computer thus avoiding conflicts. HTML being a text document rather than software is key to the success of the World Wide Web (WWW) because there is no need to install software for different sites (there are exceptions with web pages requiring plug-ins and players loaded which can be of great frustration).

Table 1 Examples of HTML tags and attributes (keywords)

Tag	Attribute	Value
Head	Value	Submit
Title	Selected	Button
Body	Border	Checkbox
BR	Type	
B	Value	
Select	SRC	
Option	Href	
Table		
TR		
TD		
Input		
Img		
A		

Web pages sometimes use JavaScript to validate user input and dynamically make text and images etc. visible or invisible depending on prior selections etc. The script is interpreted in a secure "sandbox" within the browser without access to computer resources.

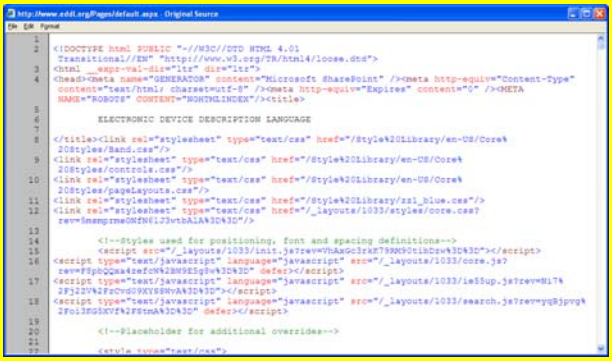

Figure 1 Plain HTML text will generate a beautiful, easy to use, graphical web page

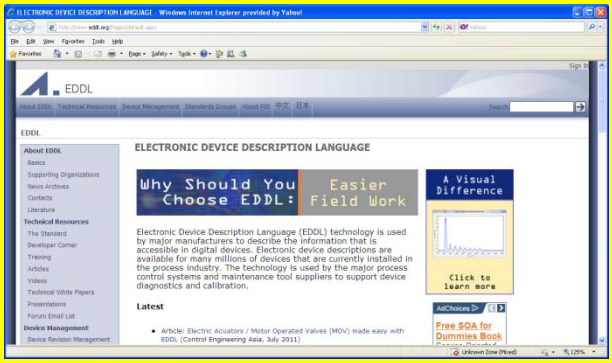

Figure 2 Beautiful, easy to use, graphical web page generated from plain HTML text

The person that surfs the web need not understand the HTML-file language. It just works.

EDDL

An EDDL file is a compressed text document, not executable software, just like an HTML file and device management software works like web browser, providing these same robustness. An EDDL

file describes the user interface content & structure the device manufacturer wants for the device. When a new device type or version arrives in the plant, the EDDL file for the device is copied onto the system, not installed. The intelligent device management software renders the graphics for configuration/setup, calibration, and diagnostics pages.

Table 2 Examples of EDDL tags and attributes (keywords)

Device Definition	Business Logic	User Interface Description	Attribute
BLOCK VARIABLE	METHOD IF SELECT * / + - FILE	MENU WAVEFORM CHART GAUGE GRAPH GRID	LABEL HELP CLASS HANDLING TYPE VALIDITY

User guidance wizards (EDDL methods) are created by the device manufacturer using a JavaScript-like language part of the EDDL standard. Wizards are interpreted by the device management software.

```
VARIABLE pressureValue
{
    LABEL [pressure_value];
    HELP [digital_value_pressure_help];
    CLASS CORRECTION & DYNAMIC;
    HANDLING READ;
    TYPE FLOAT
    {
        DISPLAY_FORMAT ".3f";
    }
}

VARIABLE pressureUnits
{
    LABEL [pressure_value_unit];
    HELP [digital_units_pressure_help];
    HANDLING READ & WRITE;
    TYPE ENUMERATED (2)
    {
        // These 16-bit enumerations may be found in Common Tables, Table 2.65 (0x41)
        { 0x4101,   [InH2O],                [inches_of_water_68_degrees_F_help] },
        { 0x4102,   [InHg],                 [inches_of_mercury_0_degrees_C_help] },
        { 0x4103,   [FtH2O],                [feet_of_water_68_degrees_F_help] },
        { 0x4104,   [mmH2O],                [millimeters_of_water_68_degrees_F_help]
```

Figure 3 Plain EDDL text will generate a beautiful, easy to use, graphical device page

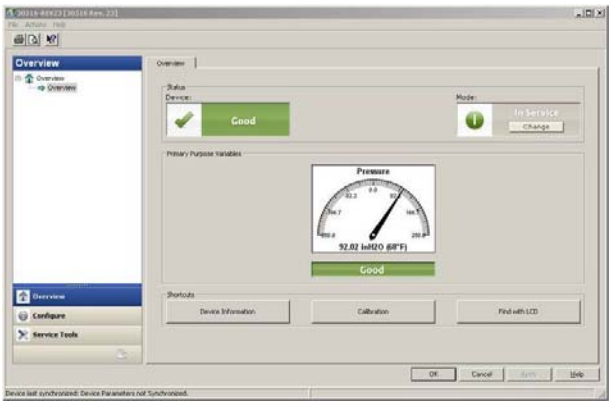

Figure 4 Graphical device page generated from plain text EDDL

The person that 'surfs' the devices need not understand the EDDL-file language. It just works.

Since EDDL files are compressed text documents, not executable software, they are simply copied and pasted onto the system. There is no installation of software. Therefore system software components are not overwritten and no Windows registry changes made. EDDL is non-interfering. Therefore the system remains robust even as EDDL files for more and more device types and versions are added over the years. Another important aspect is that the EDDL files are stored in different folders for each protocol, manufacturer, and device type. Each version of the device has a file with a different name. This scheme ensures an EDDL file for one device does not overwrite that of another. This again ensures robustness of the system. It is also possible, and usual practice, to load new EDDL devices one by one, not a whole family

At the time of intelligent device management software upgrade, only the single ‘monolithic’ software, tested by the manufacturer as a whole, is upgraded which carries less risk than installing many new driver software programs from different manufacturers one by one.

System Robustness

EDDL is the only device integration technology that works like web browsers. Other device integration technologies cannot achieve comparable results.

References

EDDL Brochure and Technical Description on www.eddl.org site

Jonas Berge, "Fieldbuses for Process Control: Engineering, Operation, and Maintenance", ISA, 2002, ISBN 1-55617-760-7

"Workshop Geräteintegration - Geräteintegration aus Anwendersicht" (Device Integration Workshop - Device Integration from a User Perspective), NAMUR AK 2.6 – Feldbus, 15. April 2003, Martin Schwibach, BASF AG Ludwigshafen

NAMUR NE 105 "Specifications for Integrating Fieldbus Devices in Engineering Tools for Field Devices", 23.09.2008

"EDD und FDT/DTM - Wo liegen die Unterschiede?" ("EDD and FDT/DTM - Where is the difference?"), NAMUR General Assembly 2005, 11 November 2005, Martin Schwibach, BASF & Michael Pelz, Clariant

"EDDL (Electronic Device Description Language)"- Podcast - Automation Minutes 10.28.08

"FDT/DTM or EDDL for asset management using FF technology", Test Report: T 2768 X 07 Published by WIB, November 2007

"Nutzung von DTMs in Prozessleitsystemen" (Use of DTMs in Process control systems) IGR Empfehlung Nr. (Recommendation No.) 04/2008/EMR, 22.08.2008

"Advantages of DCS with Integrated Instrument Management", The Dow Chemical Company, Ernst Quelle, 2009