DisplayML Implementation Contract

SWARCO I First in Traffic Solutions.

TABLE OF CONTENTS

1 DisplayML an introduction

The interface between systems and signs has historically required encoded character strings, which whilst functional can lead to complexity in creation and debugging.
Swarco Mizar has developed an XML interface for this purpose. DisplayML is Swarco Mizar standard for encoding display information for display devices.
DisplayML is a standard protocol, based on XML messages, which provides a number of distinct advantages over alternative methods;

- Human readable format
- Open; XML may be used to exchange data with other users and applications in a platform-independent way
- Easy to validate using XML Schemas
- Easily extensible
- Self-describing
- Extensive tool support

DisplayML is suitable for network connections with high bandwidth. For slower connections or in cases where you pay per MB data transferred, you should consider compressing the data, or even using another protocol.

2 Terminology

The following terms are used in this document:

DisplayML	XML-based, free-to-use protocol for presentation of dynamic information, owned and developed by Swarco Mizar.
Display Device	A device used for presentation of dynamic information, e.g. a variable message sign, a VGA monitor, an audio system or a traffic light controller.
DisplayML Implementation	A project or product in which DisplayML is implemented. A project might contain several DisplayML Implementations, e.g. if there are several types of Display Devices supporting different parts of DisplayML. In this case, a separate DisplayML Implementation Contract is required for each type of Display Device.
Information Provider	The organization or application responsible for putting together and providing dynamic information in a specific DisplayML Implementation.
Information Receiver	The organisation, application or Display Device responsible for receiving and presenting dynamic information in a specific DisplayML Implementation.
DisplayML Implementation Contract	An agreement between an Information Provider and an Information Receiver, defining which parts of the DisplayML standard will be supported in a specific DisplayML Implementation.

3 Summary

DisplayML is a highly flexible protocol for dynamic presentation of information. When implementing DisplayML, it is important to understand that you do not have to support every detail of the protocol. The important thing is that you clearly define what parts of the protocol will be used in your implementation.

You also have to define some application-specific parameters and fault codes.

This document can be used as a contract between the Information Provider and the Information Receiver, in order to formalize these decisions.

This contract is intended for project-based implementation of DisplayML. If you are an information receiver and want to state that your products are fully DisplayML compatible, you have to support the entire protocol.

4 Project details

Project Name _____
Information Provider
Information Receiver
Project Description_____
_____
_____
_____
Other information about the project (deadlines etc) _____
_____
_____
_____
_____
_____
_____
_____
_____
_____

5 DisplayML features

DisplayML has a number of built-in features. However, it is not necessary for each DisplayML application to implement all of these. The table below indicates what features will be supported in this particular implementation.

Feature	Will be implemented
Remote software upgrade	❑ Yes ❑ No
Automatic font distribution	❑ Yes ❑ No
Automatic template distribution	❑ Yes ❑ No
Usage of foreground images	❑ Yes ❑ No
Usage of background images	❑ Yes ❑ No
Usage of background color	❑ Yes ❑ No
Transfer of images	❑ Yes ❑ No
Scaling of images (scalePixelWidth, -Height)	❑ Yes ❑ No
Transfer of files by reference	❑ Yes ❑ No
Transfer of software by reference	❑ Yes ❑ No
Set system parameters (setParameters)	❑ Yes ❑ No
Request current parameter settings (getParameters	❑ Yes ❑ No
Request currently displayed info (getDisplay)	❑ Yes ❑ No
Request current status (getStatus	❑ Yes ❑ No
Clock synchronisation (clockSync)	❑ Yes ❑ No
Hardware test (hardwareTest)	❑ Yes ❑ No
Region size and position expressed in characters	❑ Yes ❑ No
Region size and position expressed in pixels	❑ Yes ❑ No
Region timeout	❑ Yes ❑ No
Horizontal alignment of text (align	❑ Yes ❑ No
Vertical alignment of text (valign)	❑ Yes ❑ No
Scrolling text (scrollUsed	❑ Yes ❑ No
Alternating text	❑ Yes ❑ No
Flashing text (flashWithInterval)	❑ Yes ❑ No
Bold text	❑ Yes ❑ No
Italic text	❑ Yes ❑ No
Underlined text	❑ Yes ❑ No
Automatic resizing of text (autoResize)	❑ Yes ❑ No
Mark-up of data contents	❑ Yes ❑ No
Numbering of data records	❑ Yes ❑ No
Display current time	❑ Yes ❑ No
Display current date	❑ Yes ❑ No
Countdown to specific time	❑ Yes ❑ No
Usage of specific speech information	❑ Yes ❑ No
Speech volume adjustment	❑ Yes ❑ No

Other decisions

DisplayML version	
File transfer encoding mechanism (if used)	❑ base64 ❑ CDATA
Interpretation of overlapping regions (if used)	❑ Transparent ❑ Solid (latest on top)

6 System parameters

DisplayML allows for definition of your own system parameters for exchange of proprietary information. The following parameters will be used in this particular implementation.

Parameter	Data type	Description