



## ***IFRA Track 4.1***

# **Introduction and Overview**

Document Date: 2008-12-01

Status: Approved

# **Technical Specification**



# Contents

<b>1</b>	<b>IFRA Track Standard Documentation</b>	<b>3</b>
1.1	Document Status and Copyright.	3
1.2	Purpose and Audience	3
1.3	Accompanying documents	3
1.4	Definitions of key words used in the specification	3
1.5	Document Revision History	3
1.6	Acknowledgements	3
1.7	Known Issues	3
<b>2</b>	<b>IFRA Track 4.1 Overview</b>	<b>3</b>
<b>3</b>	<b>Release Notes for IFRA Track 4.1</b>	<b>3</b>
3.1	IfraTrack 4.1 General features	3
<b>4</b>	<b>IFRA Track Architecture and Technical Approach</b>	<b>3</b>
4.1	Schema Architecture and XML Implementation	3
4.1.1	Schema files and namespaces	3
4.1.2	Version and identification	3
4.1.3	Controlled Vocabularies	3
4.1.4	XML Naming Conventions	3
4.2	Globally Unique Identifiers	3
<b>5</b>	<b>Creating Interfaces - Extending the IFRA Track Object Model</b>	<b>3</b>
<b>6</b>	<b>REVISION INDEX</b>	<b>3</b>

# 1 IFRA Track Standard Documentation

## 1.1 Document Status and Copyright.

This is the Approved Specification of the IFRA Track 4.1. (NOTE: version 4.0 was not officially published).

Information in this document is made available for the public good, may be used by third parties and may be reproduced and distributed, in whole and in part, provided acknowledgement is made to IFRA and provided it is accepted that IFRA rejects any liability for any loss of revenue, business or goodwill or indirect, special, consequential, incidental or punitive damages or expense arising from use of the information.

Copyright © 2008 IFRA. All rights reserved

## 1.2 Purpose and Audience

This document provides an overview of the IFRA Track standard including technical architecture, use of XML Schemas etc.

The intended audience for this document is primarily user and vendor organizations who seek to implement the IFRA Track standard in their workflows, advertising systems, or software products. Those assessing the conformance of vendor products to the standard may also use the document.

Comments on this specification should be addressed to [ifratrack@ifra.com](mailto:ifratrack@ifra.com).

## 1.3 Accompanying documents

This document provides an overview of and an introduction to the IFRA Track standard including technical architecture, use of XML Schemas etc. Detailed specifications of the object model [1] and the XML message format [2] are also available.

Descriptions of the earlier version IFRA Track 3.0 published as IFRA Special Reports [3] includes sections that provide industrial background, scope and context. These sections are still valid for the new version 4.1.

[1] IFRA Track 4.1 Object Classes, 2008

[2] IFRA Track 4.1 Message Format, 2008

[3] IFRA Track 3.0, Special Report 6.21.3, 2002

## 1.4 Definitions of key words used in the specification

The key words "**MUST**", "**MUST NOT**", "**REQUIRED**", "**SHALL**", "**SHALL NOT**", "**SHOULD**", "**SHOULD NOT**", "**RECOMMENDED**", "**MAY**", and "**OPTIONAL**" in this document are used as described in IETF RFC 2119.(S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*. Internet Engineering Task Force (IETF), Request for Comments: 2119, March 1997, <http://www.ietf.org/rfc/rfc2119.txt>)

When any of these words do not appear in upper case as above, then they are being used with their usual English language sense and meaning.

## 1.5 Document Revision History

See list of a revisions at the end of this document.

## 1.6 Acknowledgements

IFRA Track 4.1 and this document is a product of IFRA. It is based on a proposal from PRIME NETWORK ([www.prime-network.com](http://www.prime-network.com)), an association of suppliers of newspaper production systems.

IFRA Track 4.1 is heavily based on earlier versions of the standard, and IFRA thanks all contributors that have participated in the development of IFRA Track since its origin back in 1994.

Primary authorship and editing of this document was performed by:

- Ulf Wingstedt (CNet Svenska)

IFRA also thanks the AdsML Consortium ([www.adsmml.org](http://www.adsmml.org)) whose technical approach and documentation model have served as inspiration for IFRA Track.

## 1.7 Known Issues

There are no known issues in this document.

## 2 IFRA Track 4.1 Overview

The primary objective of IFRA Track is to define a way of exchanging status and management information between newspaper production management systems within a newspaper production site. The mechanisms and information structures defined in the IFRA Track specifications will make it possible to create an overall unified newspaper production tracking and planning system that can collect information and communicate with various specialised production systems supporting different parts of the newspaper production process.

The IFRA Track specifications define what is often called a *framework*, a common base for implementation of specific solutions. IFRA Track defines a large set of basic objects that serves particular purposes in the production process, for instance, a PrintingJob object for tracking purposes including properties such as number of copies printed, waste and speed. Also, IFRA Track specifies how these objects should be assembled and packaged into an XML message, the IFRA Track Message Format.

IFRA Track objects from the framework are intended to be used by production system suppliers in implementation of so called *interfaces*. An interface defines use of a subset of the IFRA Track objects which also may be extended with additional interface specific properties and controlled vocabularies (i.e. lists of values).

## 3 Release Notes for IFRA Track 4.1

Based on experiences from IFRA Track 3.0, released in 2002, the Prime Network ([www.printmedianetwork.net](http://www.printmedianetwork.net)) and IFRA have joined forces to create a new version where object classes have been extended and redefined to include properties used in practice. A draft version 4.0 was developed by PRIME but never released for public use. This version, IFRA Track 4.1, is then the official release that will replace all earlier versions of the standard.

IFRA Track 4.1 is implemented based on XML Schema technology enabling full schema based validation of structure and content of IFRA Track messages. It has been implemented to enable schema based extension of base objects in interfaces using simple but yet powerful methods.

Although the message structure of the new version is similar to the older 3.0 version, it should be noted that IFRA Track 4.1 is **not** backwards compatible with earlier versions.

### 3.1 IfraTrack 4.1 General features

- Support of planning and tracking object classes
  - Object classes extended / redefined with object classes used in practice
- Extensible object classes
  - Possible to extend the object classes within specific interfaces, including interface specific definitions of value lists (controlled vocabularies)
  - Flexible use of object class properties
  - Object class properties are not restricted to simple types
- Simplified message format
  - Link Actions removed
  - Simplification of object actions (put and purge)
  - Interface id, name and version added to message header
  - Simplification of the object class predefined structure
- Fully validatable XML files
  - All object classes with properties have been modelled in XML Schema
- Support of shared definitions for planning and tracking object class properties

# 4 IFRA Track Architecture and Technical Approach

IFRA Track specifies a set of object classes with properties. They are divided into two main groups, planning and tracking objects. Common components that are reused by several object classes are part of the Shared Definitions type library. The IFRA Track object model is described in detail in [1].

The IFRA Track Message Format is an XML format designed to be used as a wrapper for packages of IFRA Track objects. It is described in [2].

## 4.1 Schema Architecture and XML Implementation

The IFRA Track object model has been implemented as a set of XML Schema definitions divided into several schema files. Each schema file defines components in a specific target namespace.

### 4.1.1 Schema files and namespaces

- IfraTrack-4.1-PlanningObjects-AS.xsd
  - Defines the XML implementation of Planning objects. It imports the Shared Definitions, Message Format and Controlled Vocabularies schemas.
  - Namespace: <http://www.ifra.com/IfraTrack/ipo/4.1>
  - Namespace Prefix: ipo
- IfraTrack-4.1-TrackingObjects-AS.xsd
  - Defines the XML implementation of Tracking objects. It imports the Shared Definitions, Message Format and Controlled Vocabularies schemas.
  - Namespace: <http://www.ifra.com/IfraTrack/ito/4.1>
  - Namespace Prefix: ito
- IfraTrack-4.1-SharedDefinitions-AS.xsd
  - XML Definitions of reusable common components. It imports the Controlled Vocabularies schema.
  - Namespace: <http://www.ifra.com/IfraTrack/isd/4.1>
  - Namespace Prefix: isd
- IfraTrack-4.1-MessageFormat-AS.xsd
  - The IFRA Track Message Format defines the XML transport package, independently of the object model.
  - Namespace: <http://www.ifra.com/IfraTrack/imf/4.1>
  - Namespace Prefix: imf
- IfraTrack-4.1-ControlledVocabularies-AS.xsd

- Definitions of all recommended value lists, i.e. Controlled Vocabularies. It imports the Shared Definitions schema.
- Namespace: <http://www.ifra.com/IfraTrack/icv/4.1>
- Namespace Prefix: icv

#### 4.1.2 Version and identification

IFRA Track uses version numbers with three digits, where the first two are called “major version” and the third is the “minor” version. For instance, the 4.1.0 is the first minor version (0) of the 4.1 major version.

The rule governing version increments is based on if an update is backwards compatible or not. A “backwards compatible” update is an update of the standard where XML document instances of the previous version will continue to be valid under the new version. Note however that document instances that are valid according to the new version may not be valid according to the older version.

In case an upgrade of the standard is not backwards compatible, the major version number MUST be incremented. For instance, a non-backwards compatible upgrade of version 4.1.0 would lead to an increment of any of the first two digits, 5.0.0, 4.2.0 or even 4.5.0 would all be valid version numbers.

The major version number is reflected in schema file names and in all namespace definitions. This implies that a change of the major version number will lead to change in namespace definition.

The complete version number using three digits is used inside each schema file, recorded in the version attribute of the root xs:schema element. Note that all schema files have the same version, i.e. an upgrade may be performed in only one of the schema files but still resulting in a version increment for all schema files.

Each schema file includes also additional metadata recorded as attributes in the xs:schema root element that can be used to identify the schema file:

- **Status:** The current status of the schema is recorded in the isx:status attribute. Released official schemas will have the status ‘ApprovedSpecification’ (AS). Schemas released for evaluation will have the status “ProposedSpecification” (PS) while working material is labelled as “WorkingDraft” (WD).
  - Note: The status acronym is also used in the schema file names.
- **Internal draft version:** Each schema file has a draft number that is incremented independently. The draft number is reset to 1 each time a major version increment is done.
- **Identifier.** An id such as ‘IfraTrack-4.1.0-PlanningObjects-AS-5’ which is made from a combination of other schema metadata can be used as a unique identifier for the file.

#### 4.1.3 Controlled Vocabularies

The term Controlled Vocabularies, shortened CV, refers to managed list of values that may be used in specific context. For instance, the CV PageFormat may include values ‘broadsheet’, ‘tabloid’, and ‘halftabloid’.

In order to facilitate implementation while maintaining interoperability, implementers of IFRA Track interfaces can define their own CV that can replace the built in IFRA



Track CVs. A number of elements in IFRA Track has been defined on abstract types where users may either choose to use the built in IFRA Track types, or a type defined in the Interface schema.

For instance, consider the ipo:Book element with the following definition:

```
<xs:complexType name="Book">
  <xs:complexContent>
    <xs:extension base="isd:AbstractObject">
      <xs:sequence>
        <xs:element name="BookPos" type="xs:positiveInteger"/>
        <xs:element name="NoOfPage" type="xs:positiveInteger"/>
        <xs:element name="PageFormat"
          type="icv:AbstractPageFormatType"/>
        <xs:element name="Stitched" type="xs:boolean"
          minOccurs="0"/>
        <xs:element name="ProductLink" type="isd:LinkType"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Here, the PageFormat element is defined as an icv:AbstractPageFormatType. The namespace prefix icv specifies that the type definition is located in the CV schema:

```
<xs:complexType name="AbstractPageFormatType" abstract="true">
  <xs:simpleContent>
    <xs:extension base="xs:string"/>
  </xs:simpleContent>
</xs:complexType>
```

The icv:AbstractPageFormatType is based on a simple xs:string type, but defined as “abstract” which makes it mandatory to provide a type definition in an XML document instance. In all cases where an abstract CV types is being used in element definitions in IFRA Track, a built in CV is also available. In this case, the icv:PageFormatType:

```
<xs:complexType name="PageFormatType">
  <xs:simpleContent>
    <xs:restriction base="AbstractPageFormatType">
      <xs:enumeration value="broadsheet"/>
      <xs:enumeration value="tabloid"/>
      <xs:enumeration value="halftabloid"/>
    </xs:restriction>
  </xs:simpleContent>
</xs:complexType>
```

The PageFormatType is defined as a restriction of the abstract parent type, listing three values. This type may then be used in a document instance by the xsi:type definition mechanism, e.g:

```
<ipo:PageFormat
  xsi:type="icv:PageFormatType">broadsheet</ipo:PageFormat>
```

In specific Interface specifications, implementers may provide additional CV type definitions that if based on the same abstract base type, can be used instead of the IFRA Track built in CV types.

#### 4.1.4 XML Naming Conventions

All element, attribute, and type names follow the 'CamelCase' convention.

Element and type names begin using upper camel case and begin with capitals (UpperCamelCase). For example, 'PrintingJob', and 'PlateSizePosition'.

Attribute names begin using lower camel case and begin with lower case (*lowerCamelCase*). For example, 'supplier' or 'application'.

## 4.2 Globally Unique Identifiers

IFRA Track requires use of globally unique identifiers (GUID) in many locations in a XML message. It lies within the responsibility of the application creating an object instance to ensure that the identifier given to the object is globally unique, i.e. that the object can coexist with objects of the same type created by other applications without any risk of the same id being used for identification of other different objects.

The application creating a GUID is free to do so using any available mechanism and format that may be available in the technical infrastructure of the application. It is however **RECOMMENDED** to create a GUID using the following scheme:

PrintingCompany.com:2008-01-01:MyApplication:1234

The '1234' is the local id as used within the application. The 'MyApplication' is the name of the application, which must be unique among the application names from the supplier, which in turn is identified by its domain name 'PrintingCompany.com' and '2008-01-01', a date when this domain name was registered to the application supplier. All parts concatenated and separated by colon will be a valid GUID.

For example:

```
<isd:ObjectUID>PrintingCompany.com:2008-01-01:Print:11</isd:ObjectUID>
```

## 5 Creating Interfaces - Extending the IFRA Track Object Model

IFRA Track objects **MUST** be used within an Interface. Metadata identifying the Interface are **REQUIRED** in the IFRA Track Message Format header.

An Interface **MAY** include an Interface XML Schema where the IFRA Track basic objects may be extended and/or interface specific controlled vocabularies defined. In case such a schema is defined, it **MUST** be defined in a separate target namespace. The Interface schema needs to import either the planning or tracking objects schemas (or both) and also the CV schema in case Interface specific CVs are defined.

It is **RECOMMENDED** that the extended objects do not override properties or links defined in the IFRA Track standard

## 6 REVISION INDEX

Rev.	Page (P) Section (S)	Description	Involved companies	Date Dept./Initials
A	All	First approved version.	CNet, ABB, EAE, IFRA	2008-12-01/UW