

MTV Builder™ and CORBA MTV Server Generation

Chuck Plinta
Accel Software Engineering

Overview

MTV Builder is an easy to use, PC-based tool that simplifies message format specification and automates software generation to facilitate communication between systems and devices.

Users specify message formats using the GUI and the tool automatically generates the message specification reports and software. The generated software provides message translation and validation (MTV) capabilities that are easily integrated into an application. This MTV software can take several forms, all of which are based on the Ada95 programming language:

1. Ada95 MTV API (application programmer interface)
2. CORBA MTV Server
3. Database MTV Server

The generated MTV software can be ported to any hardware and operating system platform that hosts an Ada95 compiler (e.g., Object Ada™, GNAT™, APEX™, etc.). The MTV Servers also require additional products from Objective Interface Systems (OIS). CORBA MTV Servers require *ORBexpress*™ and Database MTV Servers require *Ada_SQL_Connect*™.

CORBA MTV Servers

MTV Builder generates CORBA MTV Server on a per message basis. A CORBA MTV Server is an object that provides message translation and validation services to CORBA 2.0 compliant applications. We will describe the CORBA MTV Servers generated using a simple example, the **Personnel Message**. Figure 1 shows a set of three message specification reports for the Personnel Message. The first is the Message Specification Report containing the parts (groups and fields) that are defined at the message level. The second is the Group Specification Report containing all groups that are defined within the context of the message. Each group specification contains the parts that comprise the group. The parts are shown with their properties defined within the context of the group. Finally, the third report is the Field Specification Report containing all fields that are defined within the context of the message and all its groups. The groups and fields in the reports are a subset of those in the entire message set.

From these specifications, MTV Builder generates the IDL interface package shown in Figure 2 (*Personnel_ORB.IDL*). This interface defines the IDL data types necessary to represent a Personnel Message and the services necessary to provide application developers with translation and validation capabilities. The **ToValue** function converts a formatted character-based or bit-based message string to a value based on the IDL data structure definitions. The **ToMessage** function does the opposite and converts an IDL structured data value to a formatted character-based or bit-based message string. Figure 3 shows a graphical representation of how the CORBA MTV Servers works.

Figure 1: Personnel Message - Specification Reports

Message Specification Form

Message Set Name SQL_TEST

Message Name Personnel

Message Name PERSONNEL

Ordered

| Part Name | Part Information | | | | Repeat Information | | |
|--------------|------------------|-------|-----------|----------|--------------------|-----|--------------|
| | Type | m/o/c | Delimiter | MaxBytes | Min | Max | Repeat Delim |
| 1. aName | Char Field | (m) | Slash | 20 | | | |
| 2. Education | Char Group | (m) | | 5 | | | |

Group Specification Form

Message Set Name SQL_TEST

Message Name Personnel

Group Name EDUCATION

| Part Name | Part Information | | | | Bit-Based Information | | | Repeat Information | | |
|-----------|------------------|-------|-----------|-----------------|-----------------------|-----|------|--------------------|-----|--------------|
| | Type | m/o/c | Delimiter | MaxBytes (Bits) | Byte | Bit | Mask | Min | Max | Repeat Delim |
| 1. Year | Char Field | (m) | | 4 | | | | | | |
| 2. Level | Char Field | (m) | | 1 | | | | | | |

Field Specification Form

Message Set Name SQL_TEST

Message Name Personnel

| Name | Width (Bits) | | ± | Ø | Fix | Map Type | | Representations | |
|-------|--------------|----------|--------------------------|-------------------------------------|-------------------------------------|------------|------------|----------------------------|---------------------------------------------------------------------------|
| | MinChars | MaxChars | | | | External | Internal | External | Internal |
| aName | 3 | 20 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | Alphabetic | «» String | | UnBoundedString(20) |
| Level | 1 | 1 | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | Numeric | «» Enum | 1 2 3 4 5 6 | Pre_High_school High_School Associates Bachelor Master PhD |
| Year | 4 | 4 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | Numeric | «» Integer | 0..9999 | 0..9999 |

Message Hierarchy Form

Message Set Name SQL_TEST

Message Name Personnel

| Part Name [Tag] | Kind | Mapping | Size | Delim | Repeats | RptDelim |
|-------------------|-----------|----------------------|--------|-------|---------|----------|
| 0. —0.0 Personnel | M-byte() | [Message «»Record] | [26] | | | |
| 1. —1.1 aName | F-char(m) | [Alphabet«»String] | (3:20) | Slash | | |
| 2. —1.2 Education | G-byte(m) | [Element «»Record] | [5] | | | |
| 3. —2.1 Year | F-char(m) | [Numeric «»Integer] | (4:4) | | | |
| 4. —2.2 Level | F-char(m) | [Numeric «»Enum] | (1:1) | | | |

Figure 2: Personnel Message - CORBA IDL Interface Code

```

//MTV SERVER: PERSONNEL Message (CORBA IDL Interface)
interface Personnel_ORB {
    //FIELD: YEAR (Numeric <-> Integer)
    typedef long Year_Type;

    //FIELD: LEVEL (Numeric <-> Enum)
    enum Level_Type {High_School,Associate,Bachelor,Master,Doctorate};

    //FIELD: ANAME (Alphabetic <-> String)
    const long aName_Width = 20;
    typedef string aName_Type;

    //GROUP: EDUCATION (data structure)
    struct Education_Type {
        Year_Type      Year;
        Level_Type     Level;
    };

    //MESSAGE: PERSONNEL (data structure)
    struct Personnel_Type {
        aName_Type     aName;
        Education_Type Education;
    };

    exception TV_Error {string err_msg;};
    exception ORB_Error {string err_msg;};

    //MTV SERVER INTERFACE: PERSONNEL
    string      ToMessage (in Personnel_Type aValue)
                                raises (TV_Error, ORB_Error);
    Personnel_Type ToValue (in string aMsg)
                                raises (TV_Error, ORB_Error);
    void      Check (in string aMsg,
                    out string Error_Info,
                    out boolean Validity);
};

```

The code generated by MTV Builder is Ada95 source code, but a complete MTV Server application is generated that is CORBA 2.0 compliant. This application source code need only be compiled, linked, and run to enable the MTV Server for the specific message. That means any application written in another language with the ability to interface with CORBA server objects can access the services of the CORBA MTV Server for the specified message. CORBA language mappings exist for Ada, C, C++, COBOL, Java, and Smalltalk.

Finally, Figure 4 shows how the CORBA MTV Servers are available to applications written in other programming languages on the software bus. The IDL interface definition (see `Personnel_ORB.IDL` in Figure 2) is the key to application interoperability.

Figure 3: Personnel Message - CORBA MTV Server

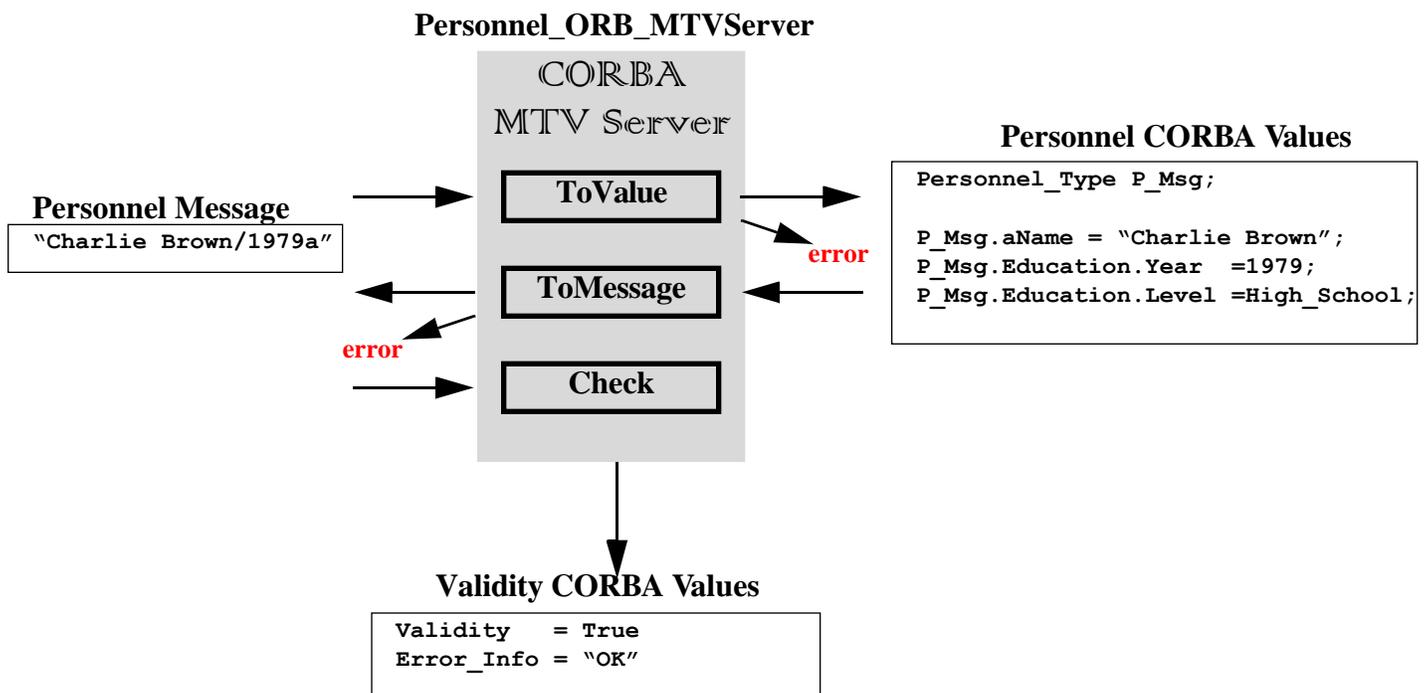


Figure 4: CORBA MTV Server Application Integration

