**Integrated Test Manager**

[**https://sourceforge.net/projects/intmac/**](https://sourceforge.net/projects/intmac/)

|  |
| --- |
|  |

Client Server Messages Specification Document

Issue V0.2.0

**Document Reference:** 04.02.02

**Document Date:** 11/02/2015

**Revision History**

|  |  |  |
| --- | --- | --- |
| Version | Date | Description |
| 0.1.0 | 08/02/2015 | Initial version for Projects List Bulk Data (Message ID 4060) |
| 0.2.0 | 11/02/2015 | * Minor typographic fixes * Font changed to Calibre to improve readability. * Added Message header section * Added Project Assets Layout Bulk Data message. * Message payloads now have an example if XML. * Document renamed to cover all messages |

**Table of Contents**

1 Introduction 1

1.1 Purpose 1

1.2 Scope 1

1.3 Referenced Documents 1

1.4 Abbreviations 1

2 Message Header 2

3 Messages 3

3.1 Projects List Bulk Data 3

3.2 Project Assets Layout Bulk Data 5

**Table of Tables**

Table 1: Referenced Documents 1

Table 2: Projects List Bulk Data Message Details 3

Table 3: Projects List Bulk Data Sources 3

Table 4: <projects> XML Tag Details 3

Table 5: < project> XML Tag Details 4

Table 6: Project Assets Layout Bulk Data Message Details 5

Table 7: Project Assets Layout Bulk Data Sources 5

Table 8: <assets> XML Tag Details 5

Table 9: <asset> XML Tag Details 6

**Table of Figures**

Figure 1: Projects List Bulk Data Body Example 3

Figure 2: Project Assets Bulk Data Body Example 5

# Introduction

The Client/Server Messages Specification Document is designed to provide documentation, which will make up the design and provide details in software development for how the message between the client and server should be structured.

## Purpose

The purpose of the specification document is to provide a detailed description of the layout of messages as pass through the communications layer across varied types of medium. The aim is to enable understanding of what the message body should consist of.

## Scope

This design document is part of the core design for the client server communications system, specifically for messages that contain bulk data, which is defined as large, structured or complex data.

## Referenced Documents

|  |  |
| --- | --- |
| Reference | Document |
| XML\_STD | <http://www.w3.org/TR/REC-xml/>  5th Edition - 26 November 2008 |

Table 1: Referenced Documents

## Abbreviations

InTMac Integrated Test Management Centre

RFC Request for Comment

XML eXtensible Markup Language

# Message Header

The first part of the message to be transmitted is the header; this details what the message and its size, giving the system flexibility for variable length message payloads.

The header is sent in a fixed format of:

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| Message ID | Unsigned integer | The unique identifier of the message. |
| Message Type | Enumeration | Defines the category of the message as:  State |
| Payload Size | Unsigned integer | The size of the messages payload (body).  A payload of 0 bytes is acceptable as some messages are triggers/events, e.g. keep-alive.  The maximum value is 512 bytes. |

# Messages

This section defines each message and it’s makeup.

## Projects List Bulk Data

The purpose of the message is to transfer a list of projects that can be accessed or selected from the server to a client. The server will filter the projects list depending on access rights and which realm a user is assigned to.

|  |  |
| --- | --- |
| Attribute | Details |
| Message ID | 4060 |
| Origin | Server |
| Destination | Client |
| Message Type | State |
| Payload Size | Variable length |
| Required Triggers | * User is Authorised * Project List Bulk has been started |
| Triggers | Project List is Loaded is triggered when the bulk data load has completed. |

Table 2: Projects List Bulk Data Message Details

### Data Input/output Sources

|  |  |
| --- | --- |
| Data Model | Data |
| State | [Input] List of selectable projects |

Table 3: Projects List Bulk Data Sources

### Message Body

The Projects Bulk Data message body is stored in XML format and consists of a list of projects, which are listed under an individual <project> XML tag.



Figure 1: Projects List Bulk Data Body Example

#### <projects> XML tag

|  |  |
| --- | --- |
| Item | Description |
| Parent | None – Root tag |
| Description | Root tag under which the projects are described. |
| Attributes | None |

Table 4: <projects> XML Tag Details

#### <project> XML tag

|  |  |  |
| --- | --- | --- |
| Item | Description | |
| Parent | project | |
| Description | Tag to describe a specific project. | |
| Attributes | ID | [Integer] - Project ID.  Validation  A value of 0 is invalid. |
| Name | [String] - Name of the project.  Validation  Minimum string length of 0 characters  Minimum string length of 150 characters  Name has to be unique (case sensitive) |
| Realm | [String] - Name of the realm hosting the project.  Validation  Minimum string length of 1 character  Minimum string length of 50 characters. |

Table 5: < project> XML Tag Details

## Project Assets Layout Bulk Data

The purpose of the message is to transfer the layout of the project assets. The server will filter the list depending on access rights, rendering some assets inaccessible.

|  |  |
| --- | --- |
| Attribute | Details |
| Message ID | 4100 |
| Origin | Server |
| Destination | Client |
| Message Type | State |
| Payload Size | Variable length |
| Required Triggers | Project Assets Requested |
| Triggers | Project Assets Loaded  Project Assets Requested is untriggered |

Table 6: Project Assets Layout Bulk Data Message Details

### Data Input/output Sources

|  |  |
| --- | --- |
| Data Model | Data |
| Project | [**TBD**] **TBD** |

Table 7: Project Assets Layout Bulk Data Sources

### Message Body

The Project Assets Bulk Data message body is stored in XML format and consists of a list of assets, which are listed under an individual <asset> XML tag.

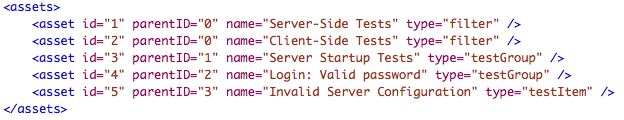


Figure 2: Project Assets Bulk Data Body Example

#### <assets> XML tag

|  |  |
| --- | --- |
| Item | Description |
| Parent | None – Root tag |
| Description | Root tag under which the assets are described. |
| Attributes | None |

Table 8: <assets> XML Tag Details

#### <asset> XML tag

|  |  |  |
| --- | --- | --- |
| Item | Description | |
| Parent | asset | |
| Description | Tag to describe a specific asset item. | |
| Attributes | ID | [Integer] – Asset ID.  Validation  A value of 0 is invalid.  ID must be unique. |
| ParentID | [Integer] – Asset ID of the parent.  Validation  A value of 0 is valid and represents root.  The parent ID must be valid. |
| Name | [String] – Name of the asset.  Validation  Minimum string length of 1 character  Minimum string length of 150 characters  Name has to be unique (case sensitive) within the scope of the filter. |
| Type | [String] – Type of assets, the following are valid (case insensitive):   * Filter – A folder that can contain filters, test items or test group. * TestItem – An individual test item. * TestGroup – A container to group tests together, it can only group TestItem. |

Table 9: <asset> XML Tag Details