



# ODYSSEY

Odyssey Protocol v3.0

Last Revision: December 4, 2014

© 2014 Atlas Systems, Inc.

Introduction.....	2
Purpose.....	2
Transport.....	2
Data Transmission.....	2
Handling Unrecognized Commands.....	2
OdysseyCommand.....	3
OdysseyStatus.....	3
OdysseyHeader.....	4
Communication.....	5
Appendix A: Odyssey Schema.....	6
Appendix B: Document Signature.....	8
Appendix C: Parameters.....	8
Appendix D: Statuses.....	9
Appendix E: Sites and Symbols.....	9
Contact Information.....	9

# Introduction

## Purpose

The Odyssey protocol was developed to transfer electronic files and metadata.

## Transport

TCP/IP on port 7968 is used as the transport mechanism.

## Data Transmission

Data is transmitted via a combination of XML data (see [Appendix A](#) for the XML schema) and file stream.

[OdysseyCommand](#), [OdysseyStatus](#), and [OdysseyHeader](#) XML data is terminated with a CRLF and followed by an empty line terminated with CRLF.

When sending a document, the stream will be read for X bytes, X being the ContentLength specified in the OdysseyHeader. The document will then be validated against the Signature specified in the OdysseyHeader. If the signature does not match, a document integrity error will be returned in the OdysseyStatus.

## Handling Unrecognized Commands

If a command is received that is unrecognized, an OdysseyStatus response should be sent using status code **710** and status message **Invalid Command**.

# OdysseyCommand

All communications begin with the transmission of an OdysseyCommand which contains the following data:

Attribute	Required	Description
protocolVersion	Yes	The version of the Odyssey Protocol being used.
version	Yes	The version of the Odyssey Command format being used.
userAgent	Yes	The client application name and optionally the version (ex. Odyssey/1.0).

Element	Required	Description
GetVersion or Put or PutMessage**	Yes	Indicates which operation to perform.
Parameters	No	Contains parameters.

\*\* Not currently supported in ILLiad

The command element (Put or PutMessage) contains the following data:

Element	Required	Description
DocId	Yes	Contains the receiver's document/message ID. This should be the same value that is set in the Receiver\DocumentID in the OdysseyHeader. If this value is not known, the value -1 may be used.

The Parameters element contains a list of Parameter elements. The Parameter element contains the value and the name attribute identifies the parameter. It is suggested to use namespaces for parameters to prevent possible overlap (ex. "atlas.citation.title" instead of "title").

# OdysseyStatus

All status messages are transmitted with an OdysseyStatus which contains the following data:

Attribute	Required	Description
implementationVersion	Yes	The maximum version of the Odyssey protocol that is supported.
protocolVersion	Yes	The version of the Odyssey Protocol being used.
version	Yes	The version of the Odyssey Status format being used.
userAgent	Yes	The client application name and optionally the version (ex. Odyssey/1.0).

Element	Required	Description
Code	Yes	Contains the status message number.
Parameters	Yes	Contains a description or more detail of the status code.

# OdysseyHeader

Document and message information is transmitted in an OdysseyHeader which contains the following data:

Element	Required	Description
Sender	Yes	Contains information about the sender.
Receiver	Yes	Contains information about the receiver.
Document	No*	Contains information on the document.
Parameters	No	Contains parameters.
Note	No	Contains a note.
Extensions	No	Contains extensions to the standard protocol.

\* Required for the Put command and for the PutMessage command to include an attachment.

The Sender and Receiver elements contain the following data:

Attribute	Required	Description
id	Yes	The Odyssey ID (ip-or-dns-name[:port][[/site]])

Element	Required	Description
Name	No	The name of the site.
Description	No	Additional information about the site.
DocumentID	No*	The site's ID for the document.

\* Required for transmission to ILLiad systems.

The Document element contains the following data:

Attribute	Required	Description
localPath	No	Typically used only on the receiving system.

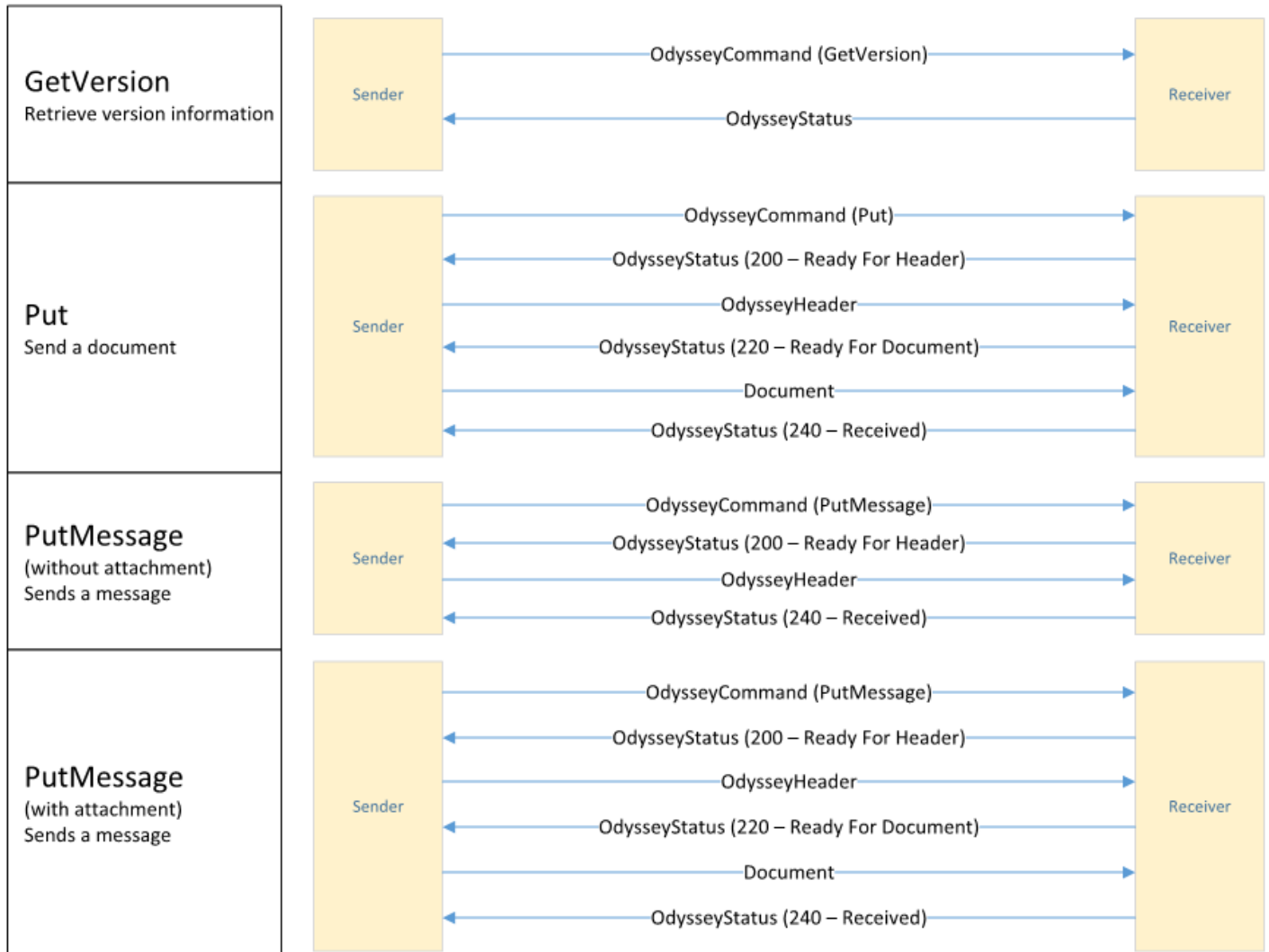
Element	Required	Description
Name	No	The name of the file.
Signature	Yes	The signature of the file which is used to ensure file validity (see <a href="#">Appendix B</a> for a detailed description of the Signature).
ContentLength	Yes	The length of the file (in bytes).
ContentType	Yes	The mime type of file (ex. image/tiff).

The Parameters element contains a list of Parameter elements. The Parameter element contains the value and the name attribute identifies the parameter. It is suggested to use namespaces for parameters to prevent possible overlap (ex. "atlas.citation.title" instead of "title"). See [Appendix C](#) for a list of currently defined parameters.

The Extensions element contains a list of Extension elements. The Extension element contains a type attribute that identifies the type of extension. It is suggested to use namespaces for extension type names to prevent possible overlap. Additional data enclosed in the Extension element can be any valid XML.

# Communication

An example of the communication that takes place for each of the different commands is outlined below.



These processes show the ideal communication between systems. Any OdysseyStatus may be different if an error occurs (document integrity error, etc.). See Appendix D for a list of the currently defined statuses.

# Appendix A: Odyssey Schema

(Also located at <http://www.atlas-sys.com/products/Odyssey/Odyssey.xsd>)

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="OdysseyCommand">
    <xs:complexType>
      <xs:sequence>
        <xs:choice>
          <xs:element name="GetVersion" type="CommandType"/>
          <xs:element name="Put" type="CommandType"/>
          <xs:element name="PutMessage" type="CommandType"/>
        </xs:choice>
        <xs:element name="Parameters" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Parameter" type="ParameterType" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attribute name="protocolVersion" type="xs:string"/>
      <xs:attribute name="version" type="xs:string" use="required"/>
      <xs:attribute name="userAgent" type="xs:string" use="required"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="OdysseyStatus">
    <xs:complexType>
      <xs:all>
        <xs:element name="Code" type="xs:int"/>
        <xs:element name="Message" type="xs:string"/>
      </xs:all>
      <xs:attribute name="implementationVersion" type="xs:string"/>
      <xs:attribute name="protocolVersion" type="xs:string" use="required"/>
      <xs:attribute name="version" type="xs:string" use="required"/>
      <xs:attribute name="userAgent" type="xs:string" use="required"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="OdysseyHeader">
    <xs:complexType>
      <xs:all>
        <xs:element name="Sender" type="SiteType"/>
        <xs:element name="Receiver" type="SiteType"/>
        <xs:element name="Document" type="DocumentType" minOccurs="0"/>
        <xs:element name="Parameters" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Parameter" type="ParameterType" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:all>
      <xs:element name="Extensions" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Extension" type="ExtensionType" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:complexType>
  </xs:element>

```

```

        </xs:element>
        <xs:element name="Note" type="xs:string" minOccurs="0"/>
    </xs:all>
    <xs:attribute name="version" type="xs:string" use="required"/>
    <xs:attribute name="userAgent" type="xs:string" use="required"/>
    <xs:attribute name="timeStamp" type="xs:string" use="optional"/>
</xs:complexType>
</xs:element>
<xs:complexType name="SiteType">
    <xs:sequence>
        <xs:element name="Name" type="xs:string" minOccurs="0"/>
        <xs:element name="Description" type="xs:string" minOccurs="0"/>
        <xs:element name="DocumentID" type="xs:string" minOccurs="0"/>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="Symbol">
            <xs:complexType>
                <xs:simpleContent>
                    <xs:extension base="xs:string">
                        <xs:attribute name="type" type="xs:string" />
                    </xs:extension>
                </xs:simpleContent>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
    <xs:attribute name="id" use="required">
        <xs:simpleType>
            <xs:restriction base="xs:string"/>
        </xs:simpleType>
    </xs:attribute>
</xs:complexType>
<xs:complexType name="CommandType">
    <xs:sequence>
        <xs:element name="DocId" type="xs:string"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="DocumentType">
    <xs:all>
        <xs:element name="Name" type="xs:string" minOccurs="0"/>
        <xs:element name="ContentLength" type="xs:int" minOccurs="0"/>
        <xs:element name="ContentType" type="xs:string" minOccurs="0"/>
        <xs:element name="Signature" type="xs:string" minOccurs="0"/>
    </xs:all>
    <xs:attribute name="localPath" type="xs:string"/>
</xs:complexType>
<xs:complexType name="ExtensionType">
    <xs:sequence>
        <xs:any namespace="##any"/>
    </xs:sequence>
    <xs:attribute name="type" type="xs:string" use="required"/>
</xs:complexType>
<xs:complexType name="ParameterType">
    <xs:simpleContent>
        <xs:extension base="xs:string">
            <xs:attribute name="name" type="xs:string" use="required"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
</xs:schema>

```

## Appendix B: Document Signature

The document signature is the hex value of the SHA1 hash that has been Base64 encoded and prepended with the string {SHA1}.

Example:

A text file containing the text Odyssey Protocol will generate a SHA1 hash as follows:

```
a15d6fecffa14bc5fde2e31a612097d044c5de58
```

If the value of the hash is Base64 encoded and prepended with {SHA1}, the resulting signature will be as follows:

```
{SHA1}oV1v7P+hS8X94uMaYSCX0ETF3lg=
```

## Appendix C: Parameters

Parameters primarily used when sending a Put command:

```
atlas.citation.journal.title  
atlas.citation.article.title  
atlas.citation.article.author  
atlas.citation.journal.volume  
atlas.citation.journal.issue  
atlas.citation.journal.month  
atlas.citation.journal.year  
atlas.citation.journal.pages  
atlas.document.pages  
atlas.patron.name  
atlas.odyssey.ill  
atlas.odyssey.note
```

Parameters primarily used when sending a PutMessage command:

```
atlas.odyssey.message.from  
atlas.odyssey.message.subject  
atlas.odyssey.message.text  
atlas.odyssey.message.id (only returned from a GetMessage command)
```

Parameters used to provide address information:

```
atlas.odyssey.address.name  
atlas.odyssey.address.line1  
atlas.odyssey.address.line2  
atlas.odyssey.address.city  
atlas.odyssey.address.state  
atlas.odyssey.address.zip  
atlas.odyssey.address.phone  
atlas.odyssey.address.email
```



## Appendix D: Statuses

OdysseyStatus codes and messages:

Code	Message
200	Ready - Send Header
220	Ready - Send Document
240	Received. Good bye.
260	Command Completed
280	OK
400	Not accepting documents.
405	Not accepting - temporary.
460	Address Unknown. Forwarding not allowed.
500	Internal Error
510	Network Error
520	Version not supported
600	Authentication Error
610	Invalid Header
615	Invalid Status
620	Wrong Document size
630	Invalid Document
640	Document Integrity Error
700	Operation Failed
710	Invalid Command
800	File Not Found

## Appendix E: Sites and Symbols

The Sender and Receiver elements, represented by the SiteType, contain the Odyssey address information for the site in the **id** attribute. This is composed of an IP or DNS name. Optionally the address may also contain the Odyssey server's port number and site code. The Odyssey port number is 7968 by default. While the site code is traditionally ILL, if a site is using an ILLiad shared server they may have a custom site code.

Symbol information may also be included in the Sender and Receiver elements. The symbol information is used to associate the site to a third party service. The current recognized symbol types are **OCLC**, **DOC** (Docline), **ISO**, and **RAPID**.

### Example

```
<Sender id="123.4.5.67:7968/ABC">  
  <DocumentID>987654</DocumentID>  
  <Symbol type="OCLC">ABC</Symbol>  
</Sender>
```

## Contact Information

Matt Calsada, Developer  
[mcalsada@atlas-sys.com](mailto:mcalsada@atlas-sys.com)

Atlas Systems, Inc.  
244 Clearfield Avenue  
Suite 407  
Virginia Beach, VA 23462  
(757) 467-7872 x208