



**Tech-XML Client/Server Version 3.0
Specification**

Production Release

August 11, 2004

License Agreement

NOTICE TO USER: CAREFULLY READ THE FOLLOWING LEGAL AGREEMENT (THE“AGREEMENT”). YOUR USE OR DOWNLOADING OF ANY SOFTWARE, TOOLS, SPECIFICATIONS OR DOCUMENTATION (COLLECTIVELY, THE “MATERIALS”) MADE AVAILABLE TO YOU BY THE MACHINERY INFORMATION MANAGEMENT OPEN SYSTEMS ALLIANCE (“MIMOSA”) CONSTITUTES YOUR ACCEPTANCE OF THESE TERMS. IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT, PROMPTLY RETURN OR DESTROY THE MATERIALS.

1. License and Restrictions. MIMOSA grants to you a non-exclusive, royalty-free, perpetual license under MIMOSA’s rights in the Materials to use and reproduce the Materials for your internal use. You may not sublicense the rights granted herein without the prior written consent of MIMOSA. You agree not to engage in or encourage the reverse engineering or reverse compilation of any object code included in the Materials.

2. Ownership of Intellectual Property. You acknowledge that title and full ownership rights to the Materials, including all proprietary rights therein, will remain the exclusive property of MIMOSA and its licensors, and that you will not acquire any rights to the Materials except as expressly set forth above. You agree not to remove any product identification, copyright or other proprietary notice from the Materials.

3. Warranties. You acknowledge that the Materials have been the result of a collective effort of several MIMOSA member organizations who have donated their time and resources and, as such, MIMOSA and its licensors make no warranty or representation, express or implied, with respect to the Materials. You agree to indemnify and hold MIMOSA harmless from and against all liabilities, losses, damages, costs and expenses, including attorneys’ fees, which MIMOSA may incur or otherwise suffer as a result of your use of the Materials.

YOU ACKNOWLEDGE THAT THE MATERIALS LICENSED HEREUNDER IS PROVIDED “AS IS,” WITHOUT WARRANTY OF ANY KIND. MIMOSA SPECIFICALLY DISCLAIMS, WITHOUT LIMITATION, ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY RIGHTS.

4. Limitation of Liability. IN NO EVENT SHALL MIMOSA OR ITS LICENSORS BE LIABLE TO YOU OR ANY OTHER PERSON OR ENTITY FOR ANY SPECIAL, CONSEQUENTIAL, INCIDENTAL, INDIRECT, EXEMPLARY OR PUNITIVE DAMAGES, HOWEVER CAUSED, WHETHER FOR BREACH OF CONTRACT, TORT, NEGLIGENCE, STRICT PRODUCT LIABILITY OR OTHERWISE (INCLUDING, WITHOUT LIMITATION, DAMAGES BASED ON LOSS OF PROFITS, DATA OR BUSINESS OPPORTUNITY), AND WHETHER OR NOT SUCH PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION SHALL APPLY NOTWITHSTANDING ANY FAILURE OF ESSENTIAL PURPOSE OF ANY LIMITED REMEDY PROVIDED HEREIN.

5. Term and Termination. The term of this license is for the duration of any copyright in the Materials. You may terminate this Agreement at any time by written notice to MIMOSA. This Agreement shall terminate immediately without notice if you fail to comply with any provision of this Agreement. Upon termination you shall immediately cease using the Materials and dispose of the Materials.

6. Government Contracts. If you are an agency or instrumentality of the United States Government, the Materials are “commercial computer software” and “commercial computer software documentation”, and pursuant to FAR 12.212 or DFARS 227.7202, and their successors, as applicable, use, reproduction and disclosure of the Materials is governed by the terms of this Agreement.

7. Export Restrictions. You acknowledge that the laws and regulations of the United States restrict the export and re-export of certain commodities and technical data of United States origin, which may include the Materials, in any medium. You agree that you will not knowingly, without prior authorization if required, export or re-export the Materials in any medium without the appropriate United States and foreign government licenses.

8. General Provisions. This Agreement shall be governed, construed and enforced in all respects by the laws of the State of California, without respect to its conflict of laws provisions. The United Nations Convention on Contracts for the International Sale of Goods is specifically excluded from application to this Agreement. In the event any action is brought for any breach or default of any of the terms of this Agreement, or otherwise in connection with this Agreement, the prevailing party shall be entitled to recover from the other party all costs and expenses (including reasonable attorneys’ fees) incurred therefrom. The relationship of the parties established by this Agreement is that of licensor and licensee, and nothing contained herein shall be construed to constitute either party as the agent of the other party as partners, joint ventures, co-owners or otherwise as participants in a joint or common undertaking. Neither party shall be liable hereunder by reason of any failure or delay in the performance of its obligations hereunder (except for the payment of money) on account of strikes, riots, insurrection, fires, flood, storm, explosions, war, governmental action, labor conditions, earthquakes, material shortage, or any other cause which is beyond the reasonable control of such party. You shall not assign or otherwise transfer any of your rights or obligations under this Agreement without MIMOSA’s prior written consent. All terms of this agreement shall be binding upon, inure to the benefit of, and be enforceable by the parties hereto and their respective heirs, legal representatives, successors, and assigns. No failure or delay on the part of either party to exercise, in whole or in part, any right or privilege hereunder shall operate as a waiver thereof or of any right to exercise or enforce such right or any other right or privilege hereunder. This Agreement constitutes the entire agreement between the parties regarding its subject matter and supersedes and its terms govern, all prior proposals, agreements or other communications between the parties regarding such subject matter. Should any part of this Agreement be held to be invalid by a court of competent jurisdiction, the remainder of this Agreement shall be considered as the whole and be binding on the parties. Should you have any questions concerning this Agreement, or if you desire to contact MIMOSA for any reason, please write: MIMOSA, 31882 Paseo Alto Plano, San Juan Capistrano, California, 92675-3406 (Tel: (+1-949-625-8616, Fax: +1-949-625-8616)

Tech-XML Client/Server Version 3.0 Specifications

August 11, 2004

A key component of MIMOSA's Open System Architecture for Enterprise Application Integration (OSA-EAI) is the *Tech-XML* Client/Server schema. These XML schema provides a common set XML-based client/server interface definitions for various protocols. The *Tech-Web* Client/Server specifications implement the *Tech-XML* schema using standard HTTP protocol, but other methods of client/server communication may be utilized. The specifications allows applications to have open access to retrieve and optionally store various types of MRO (maintenance, repair, and operations) information. This document contains the requirements which a *Tech-XML* Server system and corresponding application software must support.

Technology Types

A *Tech-XML* Server must support at least one or more of the following technology types which correspond to different types of information a server might manage:

Technology Type	Description
Asset Registry Management Information (REG-XML Server Interfaces) ["5000" series]	Allows retrieval of "as-designed" segment hierarchical breakdown of facility, process, and machine systems, along with the "as-installed" asset information. Also allows access to name plate and image data on individual assets and models, including component part breakdowns. Used by: - OEM Model Information Systems - Asset Registry Information Systems - Maintenance Management Systems - Piping & Instrumentation Design Systems
Reliability Management Information (REL-XML Server Interfaces) ["7000" series]	Allows retrieval of reliability study information related to a service segment, a model of an asset, or a serialized asset. Allows the retrieval of information associated with hypothetical causal networks, reliability-centered maintenance information, as well as actual failure events. Used by: - Reliability Information Systems - FMECA Analysis Systems - RCM Analysis Systems
Work/Action Management Information (WORK-XML Server Interfaces) ["8000" series]	Allows the creation and audit tracking of a new work request in a work management system for a service segment or a serialized asset. Allows the retrieval of work orders and work order steps, and actual work completed information.. Also allows the retrieval of pre-planned work packages ("solution packages"). Used by: - Maintenance Management Systems

Diagnostics / Health / Prognostics Information (DIAG-XML Server Interfaces) [“6000” series]	<p>Enables retrieval of human or "smart-agent" generated current and/or future proposed asset health states, current and/or future proposed diagnostic failure modes and casual trees, remaining useful life predictions, and recommendations. Also allows access to measurement evidence supporting the diagnoses/prognoses.</p> <p>Used by:</p> <ul style="list-style-type: none"> - Diagnostic Systems - Prognostic Systems
Trendable Scalar Data (TREND-XML Server Interfaces) [“1000” series]	<p>Enables the creation and retrieval of historical scalar measurements, abnormal data alarms, and operational event logs.</p> <p>Used by:</p> <ul style="list-style-type: none"> - Maintenance Management Systems for Usage-based PM Work Triggering - Process Data Historians - Process Condition Monitoring Systems - Operational Data Systems
Dynamic Data (DYN-XML Server Interfaces) [“2000” series]	<p>Enables the creation and retrieval of historical dynamic measurements (used with vibration and sound monitoring and including frequency spectra measurements and time waveforms), abnormal data alarms, and operational event logs.</p> <p>Used by:</p> <ul style="list-style-type: none"> - Vibration Condition Monitoring Systems - Sound Condition Monitoring Systems
Sampling Data (SAMPLE-XML Server Interfaces) [“3000” series]	<p>Enables the creation and retrieval of historical fluid, air, and solid sampling data, abnormal data alarms, and operational event logs.</p> <p>Used by:</p> <ul style="list-style-type: none"> - Oil Sampling Condition Monitoring Systems - Air Sampling Condition Monitoring Systems - Solid Sampling Condition Monitoring Systems
Binary Data (BLOB-XML Server Interfaces) [“4000” series]	<p>Enables the creation and retrieval of historical binary large objects (BLOB) measurements (used with thermography and imaging monitoring), abnormal data alarms, and operational event logs</p> <p>Used by:</p> <ul style="list-style-type: none"> - Thermographic Condition Monitoring Systems - Image Monitoring Systems

A given server may decide to support more than one set of interfaces from different technology types. For example, a maintenance management system may manage both asset registry information and work management information. The supplier may decide to build a *Tech-XML* Server which supports both REG-XML and WORK-XML interface types. One server can provide interface support for one or multiple databases or data sources, combining query data from the multiple databases together and returning the results to the client or creating new rows of data in the appropriate database. For ease of use, both the client request and the server acknowledgement definition are stored in a single XML schema file named in the following manner:

V3-0-xxxx-vvDescription.xsd (xxxx = unique schema number, vv = version number)

Each set of interfaces has a defined set of XML schema (XSD's). The first element of each schema begins with **mim_nnnn**, where the *nnnn* is the unique number for this schema. Each set of interfaces begins with the same "series" number, for example, the REG-Web interfaces are in the 5000 range. Sometimes these interfaces may be referenced as the "5000" series, which is synonymous with saying "Reg-XML" interfaces. There are 2 interfaces which are part of the "0000" series – connect and disconnect -- which are included in all packages. The "000" series also includes three XSD schema which are included by reference in all Tech-XML schema. These three schema are:

V3-0TechXMLInclude.xsd -- schema which includes the following:
V3-0MessageInclude.xsd – general Tech-XML message schema
V3-0CrisVocabInclude.xsd – entire V3.0 CRIS relational schema

A *Tech-XML* Version 3.0 interface is one of four types, which is reflected in the descriptive name given to the interface. The **connect** interface (**mim_0003** defined in **V3-0-0003-nnConnect.xsd**) is required to be the first interface sent by a client to a server. This is where a client provides the server with required database/system login/password information and establishes a session with the server. The server returns a session identifier to the client, which the client must then utilize on subsequent requests. Session-less servers may also be built. After interacting with the server, the client under normal circumstances will use the **disconnect** interface (**mim_0004**, defined in **V3-0-0004-nnDisconnect.xsd**) when all interactions with a server are completed. **Query** interfaces, which are prefaced with "query" in their description, request the server to return information in one or more "row" elements, filtered by an optional set of parameters (element "param") sent by the client. **Create** interfaces, request a server to add a new instance of a data item or "row" in a server.

Support Levels

A *Tech-XML* Server can support various levels of interfaces. With the V3.0 release, there are two support levels -- Level 1 and Level 2. Level 1 contains a basic set of interfaces, with level 2 supporting more functionality. A *Tech-XML* server must respond to all interfaces for a supported level. If a particular type of information is not supported by the underlying system, then the server must respond with an error return code of **0000000000000000-0006** which represents "Function Not Implemented".

Access Types

A *Tech-XML* Server can support one of the following three types of access:

Read-Only – supports query interfaces only

Write-Only – supports create interfaces only

Read/Write – supports both query and create interfaces

include_lc_info_columns_def: Boolean flag to indicate whether the server should, by default, transmit local columns using the optional "lc_info" element on returned CRIS table elements. This default can be overridden by the client on each query request. (0 = No, 1 = Yes)

If the connect is successful (the attribute **success** in element **status** equals "1"), a **header** element will be returned from the server, unless the server does not support sessions. In the **header** element, the attribute **session_id** must be utilized by the client on every subsequent request to the server during the session. If the connect is successful, the **status** element's attribute **success** will equal True (1) and the server also must send back one or more rows in the **DB_MIM_Interface** entity (table) which describes which MIMOSA Tech-XML technologies, support levels, and access types the server supports.

The Header Element

After a **connect** request, every subsequent client request to a server which supports sessions, must contain the **header** element. This element contains an optional **lc_info** sub-element which allows for the client to send locally-defined attribute-value pairs of additional server parameters which are additions to the MIMOSA-defined standard. Servers which do not support these additional parameters must ignore them. Required in the **header** element, a valid **session_id** must be specified by the client.

Other optional attributes of the **header** element are:

include_non_active_rows: Boolean flag to indicate whether the server should override the default set for this flag and transmit rows which have a row status type which is not active, i.e. inactive or soft-deleted. These are rows where `rstat_type_code != 1`. (0 = No, 1 = Yes)

include_CRIS_ref_data_rows: Boolean flag to indicate whether the server should override the default set for this flag and transmit MIMOSA CRIS Reference Data Library rows to the client. These rows are associated with all databases which are contained in any sites which are associated with enterprise 0 (MIMOSA). (0 = No, 1 = Yes)

include_row_info_columns: Boolean flag to indicate whether the server should override the default set for this flag and transmit the 4 information columns related to a row:

gmt_last_updated
last_upd_db_site
last_upd_db_id
rstat_type_code

on any returned CRIS table elements. (0 = No, 1 = Yes)

include_lc_info_columns: Boolean flag to indicate whether the server should override the default set for this flag and transmit local columns using the optional "lc_info" element on any returned CRIS table elements. (0 = No, 1 = Yes)

The Status Element

Must be returned by the server in the acknowledgement of every client request. The **status** element contains an optional **lc_info** sub-element which allows for the server to send locally-defined attribute-value pairs of additional parameters to the client which are additions to the MIMOSA-defined standard. If the connect is successful, the **status** element's boolean attribute **success** will equal True (1). If not, then an error has occurred. The client should then look at the attribute **message_code** for the codified error message from the server. The **message_code** is composed of three parts. The first part is a 16-character site code which is the hex representation of the 4-byte non-negative integer **enterprise_id** prefixed to the hex representation of the non-negative integer **site_id** of the site creator of the message code. Following this is a dash ("-"). Finally, there is a 4-byte non-negative integer message identifier expressed in text form with optional leading zeroes. The syntax in EBNF is:

Message-Code = Site-tag "-" Message-tag
 where Site_tag = CHAR(16) and
 Message-tag = non-negative INTEGER

Whitespace is not allowed within the message code. Commonly-used message codes have been defined by MIMOSA and must be utilized whenever possible. The format of these MIMOSA-defined messages is in the format "0000000000000000-0000". Servers can include additional state codes (which do not use the MIMOSA enterprise prefix of "0000000000000000") to handle errors which have not been defined by MIMOSA. The **message_text** provides a local language description of the error for the client to display in an error log or to the end-user. The **message_text** will be returned in the language specified by the client when connecting with the server.

Version 3.0 MIMOSA-defined error codes with their corresponding English text are:

0000000000000000-0001	"Invalid Connect String"
0000000000000000-0002	"Data Source Not Available"
0000000000000000-0003	"Data Limit Exceeded"
0000000000000000-0004	"No Privilege For Operation"
0000000000000000-0005	"Language Not Supported"
0000000000000000-0006	"Function Not Implemented"
0000000000000000-0007	"Cannot Create Entry - Primary Key Already Exists"
0000000000000000-0008	"Cannot Create Entry - Not All Required Data Specified"
0000000000000000-0009	"Cannot Create Entry - Invalid Data Specified"
0000000000000000-0010	"Cannot Create Entry - Exceeded Attachment Limit"
0000000000000000-0011	"Cannot Create Entry - Attachment Data Type Not Supported"
0000000000000000-0012	"Session ID Not Valid - Connect Required"

The optional **row_count** attribute in the **status** element returns the number of rows for queries where the **count_only** attribute in the **param** element is set to true ("1").

The Server Element

The **server** element is returned by the server in the acknowledgement of a connect request. This element contains the **name** attribute which contains the name of the server. The **server** element also contains an optional **lc_info** sub-element which allows for the server to send locally-defined attribute-value pairs of additional parameters to the client which are additions to the MIMOSA-defined standard.

Query Interfaces

In each query interface, following the **mim_nnnn** element, the client sends a **query_description_req** element. The subsequent **header** element contains the session identifier (unless session-less) and the **param** element provides attributes to filter the query to the server. The client will then wait for a client-determined interval of time for a response from the server. Assuming the server is able to respond, the server will return a **query_description_ack** element. In this element is the **header** element sent by the client (optional), the **param** element sent by the client (optional), a mandatory **status** element where the client will check for success in the **success** attribute, and an unlimited number of **row** elements where the data is returned, if the query was successful. If the client set the **count_only** attribute on the **param** element to be true, then the number of rows the server would have returned is stored in the returned **status** element in the **row_count** attribute.

The **lc_info** element is an optional part of the **param** structure and allows for the client to send locally-defined server parameters which are additions to the base MIMOSA-defined standard. Servers which do not support these additional parameters must ignore them. The **lc_info** element is also an optional part of every returned **row** structure and allows for the server to return locally-defined data which is not recognized in CRIS V3.0. Clients which do not support this additional data should ignore it.

Create Interfaces

In each create interface, following the **mim_nnnn** element, the client sends a **create_description_req** element. The subsequent **header** element contains the session identifier (unless session-less) and the **param** element provides any locally-defined server parameters. This is followed by the actual elements which are required for the server to create the desired row of information. The client will then wait for a client-determined interval of time for a response from the server. Assuming the server is able to respond, the server will return a **create_description_ack** element. In this element is the **header** element sent by the client (optional), the **param** element sent by the client (optional), and a mandatory **status** element where the client will check for success in the **success** attribute.

The **lc_info** element is an optional part of the **param** structure and allows for the client to send locally-defined server parameters which are additions to the base MIMOSA-defined standard. Servers which do not support these additional parameters must ignore them. The **lc_info** element is also an optional part of every transmitted **row** structure and allows for the client to send locally-defined data which is not recognized in CRIS V3.0. Servers which do not support this additional data should ignore it.

Additional Requirements for Tech-XML Server Applications

A *Tech-XML* Server application must also meet the requirements specified in the **OSA-EAI V3.0 Registry Compliance Requirements For All Producer-Export-Server Software** specification.