



OSA-EAI
Open System Architecture For
Enterprise Application Integration

Tech-Web Server Version 2.2 Specifications **(Interfaces #962XX22)** **April 26, 2003**

A key component of MIMOSA's Open System Architecture for Enterprise Application Integration (OSA-EAI) is the *Tech-XML* Client/Server schema. This 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 HTTP protocol. The specifications allows Web 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-Web* Server system and corresponding application software must support.

Server-Supported Technologies

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

Application Technology Packages	Description
Asset Register Management Information (REG-Web Server Interfaces -- #9620522) ["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: <ul style="list-style-type: none">- OEM Model Information Systems- Asset Registry Information Systems- Maintenance Management Systems- Piping & Instrumentation Design Systems
Work/Action Management Information (WORK-Web Server Interfaces -- #9620822) ["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: <ul style="list-style-type: none">- Maintenance Management Systems
Diagnostics / Health / Prognostics Information	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. Used by:

(DIAG-Web Server Interfaces -- #9620622) [“6000” series]	- Diagnostic Systems - Prognostic Systems
Trendable Process Data (TREND-Web Server Interfaces -- #9620122) [“1000” series]	Enables the creation and retrieval of historical scalar measurements, abnormal data alarms, and operational event logs. Used by: - Process Data Historians - Process Condition Monitoring Systems - Used by Operational Data Systems
Dynamic Data (DYN-Web Server Interfaces -- #9620222) [“2000” series]	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. Used by: - Vibration Condition Monitoring Systems - Sound Condition Monitoring Systems
Sampling Data (SAMPLE-Web Server Interfaces -- #9620322) [“3000” series]	Enables the creation and retrieval of historical fluid, air, and solid sampling data, abnormal data alarms, and operational event logs. Used by: - Oil Sampling Condition Monitoring Systems - Air Sampling Condition Monitoring Systems - Solid Sampling Condition Monitoring Systems
Binary Data (BLOB-Web Server Interfaces -- #9620422) [“4000” series]	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 Used by: - Thermographic Condition Monitoring Systems - Image Monitoring Systems

A given server may decide to support more than one set of interface packages. For example, a maintenance management system may manage both asset registry information and work management information. The supplier may decide to build a *Tech-Web* Server which supports both REG-Web and WORK-Web interface packages. One server can provide interface support for 1 MIMOSA CRIS (Common Relational Information Schema) database or can provide support for multiple databases, combining query data from the multiple database together “on the fly” and returning the results to the client or creating a new row of data in the appropriate database.

Each set of interfaces has a defined set of XML schema (XSD’s). The first element of each schema begins with “mim_v2_2_xxxx”, where the “xxxx” 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-Web” interfaces. There are 2 interfaces which are part of the “0000” series – connect and disconnect -- which are included in all packages.

A Tech-Web Version 2.2 interface is one of four types, which is reflected in the descriptive name given to the interface. The **connect** interface (“mim_v2_2_0003” defined in “V2-2-0003-01Connect.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. (Note: Session-less servers can be built, but are normally not recommended for robust applications.) After interacting with the server, the client under normal circumstances will use the **disconnect** interface (“mim_v2_2_0004”, defined in “V2-2-0004-01Disconnect.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.

A server must technically respond to all interfaces for a supported package, but does not have to implement the desired functionality. For every acknowledgement by a server, a “status” element is returned which has an associated “success” boolean attribute where a server can respond with reasons for unsuccessful requests. For example, the return code 0-0006 represents "Function Not Implemented" and should be sent by a server to a client if the particular interface is not supported.

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:

V2-2-xxxx-vvDescription.xsd (xxxx = unique schema number, vv = version number)

The client is required to make an **HTTP Post** request of the server with the following syntax:

Schema=*SchemaFileName* XML=*XMLstring*

In *XMLstring*, the client sends a “Description_req” element following the “mim_v2_2_xxxx” element and subsequent elements to connect to a server, filter a query to the server, or send data to create a new item in the server. The client will then wait for a response from the server. The server is required to respond with an **HTTP Post** acknowledgement with the following syntax:

Schema=*schema file name* XML=*xml string*

In *XMLstring*, the server responds with a “Description_ack” element following the “mim_v2_2_xxxx” element and subsequent elements to provide status information, a count of the rows of requested data or the actual requested data from a server.

Tech-Web Server Database Requirements

The supplier of the databases supported by the *Tech-Web Server* software must provide the end-user with the ability to configure and maintain MIMOSA/Site-assigned globally-unique identifiers and user-defined names. The supplier may provide an external software utility to perform this function. The fields the user must have the ability to assign/modify directly are:

Site.site_code

Site.company_name

Site.site_name

Site_Database.db_site

Site_Database.db_id

Site_Database.name

Agent.org_agent_site

Agent.agent_id

Agent.agent_db_site, agent_db_id, agent_type_code (via Agent_Type.name lookup)

Agent.name

Segment.segment_site

Segment.segment_id

Segment.sg_db_site, sg_db_id, sg_type_code (via Segment_Type.name lookup)

Segment.user_tag_ident

Asset.asset_org_site

Asset.asset_id

Asset.as_db_site, as_db_id, as_type_code (via Asset_Type.name lookup)

Asset.user_tag_ident

Meas_Location.meas_loc_site

Meas_Location.meas_loc_id

Meas_Location.ml_db_site, ml_db_id, ml_type_code
(via Meas_Loc_Type.name lookup)

Meas_Location.user_tag_ident

When generating new rows in the Agent, Segment, Asset, and Meas_Location tables, the end-user must be given the ability to provide to the system a site-unique range of integer identifiers (agent_id, segment_id, asset_id, and meas_loc_id, respectively) that the database may draw from when creating new entries.