



جمهوری اسلامی ایران
Islamic Republic of Iran
سازمان ملی استاندارد ایران

Iranian National Standardization Organization



استاندارد ملی ایران

۱۶۸۹۷-۳

چاپ اول

آذر ۱۳۹۲

INSO

16897-3

1st. Edition

Nov.2013

فناوری اطلاعات - تعامل پذیری و انقیادهای
مراکز ثبت فراداده (MDR-IB) - قسمت ۳:
انقیادهای واسط برنامه نویسی کاربردی (API)

Information technology - Metadata
Registries Interoperability and Bindings
(MDR-IB) - part 3: API bindings

ICS:35.040

به نام خدا

آشنایی با سازمان ملی استاندارد ایران

مؤسسه استاندارد و تحقیقات صنعتی ایران به موجب بند یک ماده ۳ قانون اصلاح قوانین و مقررات مؤسسه استاندارد و تحقیقات صنعتی ایران، مصوب بهمن ماه ۱۳۷۱ تنها مرجع رسمی کشور است که وظیفه تعیین، تدوین و نشر استانداردهای ملی (رسمی) ایران را به عهده دارد.

نام موسسه استاندارد و تحقیقات صنعتی ایران به موجب یکصد و پنجاه و دومین جلسه شورای عالی اداری مورخ ۹۰/۶/۲۹ به سازمان ملی استاندارد ایران تغییر و طی نامه شماره ۲۰۶/۳۵۸۳۸ مورخ ۹۰/۷/۲۴ جهت اجرا ابلاغ شده است.

تدوین استاندارد در حوزه های مختلف در کمیسیون های فنی مرکب از کارشناسان سازمان، صاحب نظران مراکز و مؤسسات علمی، پژوهشی، تولیدی و اقتصادی آگاه و مرتبط انجام می شود و کوششی همگام با مصالح ملی و با توجه به شرایط تولیدی، فناوری و تجاری است که از مشارکت آگاهانه و منصفانه صاحبان حق و نفع، شامل تولیدکنندگان، مصرف کنندگان، صادرکنندگان و وارد کنندگان، مراکز علمی و تخصصی، نهادها، سازمان های دولتی و غیر دولتی حاصل می شود. پیش نویس استانداردهای ملی ایران برای نظرخواهی به مراجع ذی نفع و اعضای کمیسیون های فنی مربوط ارسال می شود و پس از دریافت نظرها و پیشنهادهای در کمیته ملی مرتبط با آن رشته طرح و در صورت تصویب به عنوان استاندارد ملی (رسمی) ایران چاپ و منتشر می شود.

پیش نویس استانداردهایی که مؤسسات و سازمان های علاقه مند و ذی صلاح نیز با رعایت ضوابط تعیین شده تهیه می کنند در کمیته ملی طرح و بررسی و در صورت تصویب، به عنوان استاندارد ملی ایران چاپ و منتشر می شود. بدین ترتیب، استانداردهایی ملی تلقی می شوند که بر اساس مفاد نوشته شده در استاندارد ملی ایران شماره ۵ تدوین و در کمیته ملی استاندارد مربوط که سازمان ملی استاندارد ایران تشکیل می دهد به تصویب رسیده باشد.

سازمان ملی استاندارد ایران از اعضای اصلی سازمان بین المللی استاندارد (ISO)^۱، کمیسیون بین المللی الکتروتکنیک (IEC)^۲ و سازمان بین المللی اندازه شناسی قانونی (OIML)^۳ است و به عنوان تنها رابط^۴ کمیسیون کدکس غذایی (CAC)^۵ در کشور فعالیت می کند. در تدوین استانداردهای ملی ایران ضمن توجه به شرایط کلی و نیازمندی های خاص کشور، از آخرین پیشرفت های علمی، فنی و صنعتی جهان و استانداردهای بین المللی بهره گیری می شود.

سازمان ملی استاندارد ایران می تواند با رعایت موازین پیش بینی شده در قانون، برای حمایت از مصرف کنندگان، حفظ سلامت و ایمنی فردی و عمومی، حصول اطمینان از کیفیت محصولات و ملاحظات زیست محیطی و اقتصادی، اجرای بعضی از استانداردهای ملی ایران را برای محصولات تولیدی داخل کشور و/یا اقلام وارداتی، با تصویب شورای عالی استاندارد، اجباری نماید. سازمان می تواند به منظور حفظ بازارهای بین المللی برای محصولات کشور، اجرای استاندارد کالاهای صادراتی و درجه بندی آن را اجباری نماید. همچنین برای اطمینان بخشیدن به استفاده کنندگان از خدمات سازمان ها و مؤسسات فعال در زمینه مشاوره، آموزش، بازرسی، ممیزی و صدور گواهی سیستم های مدیریت کیفیت و مدیریت زیست-محیطی، آزمایشگاه ها و مراکز کالیبراسیون (واسنجی) و وسایل سنجش، سازمان ملی استاندارد ایران این گونه سازمان ها و مؤسسات را بر اساس ضوابط نظام تأیید صلاحیت ایران ارزیابی می کند و در صورت احراز شرایط لازم، گواهینامه تأیید صلاحیت به آن ها اعطا و بر عملکرد آن ها نظارت می کند. ترویج دستگاه بین المللی یکاها، کالیبراسیون (واسنجی) و وسایل سنجش، تعیین عیار فلزات گرانبها و انجام تحقیقات کاربردی برای ارتقای سطح استانداردهای ملی ایران از دیگر وظایف این سازمان است.

1- International Organization for Standardization

2 - International Electrotechnical Commission

3- International Organization of Legal Metrology (Organisation Internationale de Metrologie Legale)

4 - Contact point

5 - Codex Alimentarius Commission

کمیسیون فنی تدوین استاندارد

« فناوری اطلاعات - تعامل پذیری و انقیادهای مراکز ثبت فراداده (MDR-IB) - قسمت ۳:

انقیادهای واسط برنامه‌نویسی کاربردی (API) »

رئیس:

مشرف، بهنوش

(فوق لیسانس مهندسی فناوری اطلاعات- شبکه‌های کامپیوتری)

سمت و / یا نمایندگی

کارشناس استاندارد

دبیر:

ترابی، مهرنوش

(فوق لیسانس مهندسی فناوری اطلاعات - تجارت الکترونیک)

کارشناس استاندارد

اعضاء: (اسامی به ترتیب حروف الفبا)

ذاکری، صفورا

(فوق لیسانس مهندسی کامپیوتر - نرم‌افزار)

عضو هیات علمی دانشگاه آزاد اسلامی

بندرعباس

زمانی، کرشنا

(فوق لیسانس مهندسی فناوری اطلاعات - تجارت الکترونیک)

کارشناس مرکز رایانه دانشگاه مازندران

سایانی، احمد

(فوق لیسانس مهندسی برق - قدرت)

مدیر دفتر تحقیقات و استانداردها شرکت

برق منطقه‌ای هرمزگان

صحراگرد، جلیل

(فوق لیسانس مهندسی برق - قدرت)

معاون دفتر برنامه‌ریزی و تحقیقات شرکت

برق منطقه‌ای هرمزگان

فهرست مندرجات

صفحه	عنوان
ب	آشنایی با سازمان ملی استاندارد
ج	کمیسیون فنی تدوین استاندارد پیش گفتار
۱	۱ هدف و دامنه کاربرد
۱	۲ مراجع الزامی
۱	۳ اصطلاحات و تعاریف
۲	۴ موارد استفاده این استاندارد
۲	۵ مدل انتزاعی
۲	۱-۵ کلیات
۲	۲-۵ الگوی جلسه
۳	۳-۵ چارچوب امنیت
۴	۴-۵ مسیرهای پیمایش سلسله مراتبی
۴	۶ خدماتها
۴	۱-۶ استفاده از استاندارد ISO/IEC 13886
۴	۲-۶ خدماتهای برقراری جلسه
۸	۳-۶ خدماتهای پارامتر جلسه
۱۰	۴-۶ خدماتهای آمینت
۱۲	۵-۶ خدماتهای انتقال داده
۲۴	۶-۶ متفرقه
۲۹	۷ انقیادها
۲۹	۸ اداره
۲۹	۹ انطباق
۲۹	۱-۹ نمونه انطباق API
۳۰	۲-۹ برنامه کاربردی API
۳۰	۳-۹ محیط API
۳۱	۴-۹ برچسبهای انطباق

ادامه فهرست مندرجات

صفحه		عنوان
۳۱	محفوظ برای استاندارد سازی آینده	۱۰
۳۱	انقیاد C API	۱۱
۳۱	نگاشت نوع داده	۱-۱۱
۳۱	نگاشت پارامتر تابع	۲-۱۱
۳۲	نگاشت برگشت تابع	۳-۱۱
۳۲	نگاشت استثنا تابع	۴-۱۱
۳۲	امضاهای فراخوانی رویه	۵-۱۱
۴۰	خطا/استثنا برگشتی	۶-۱۱
۴۰	پیشوند برچسب انطباق	۷-۱۱
۴۱	انقیاد Java API	۱۲
۴۱	نگاشت نوع داده	۱-۱۲
۴۱	نگاشت پارامتر تابع	۲-۱۲
۴۱	نگاشت بازگشت تابع	۳-۱۲
۴۱	نگاشت استثنا تابع	۴-۱۲
۴۲	امضاهای فراخوانی رویه	۵-۱۲
۴۸	خطا/استثنا برگشتی	۶-۱۲
۴۸	پیشوند برچسب انطباق	۷-۱۲
۴۸	انقیاد ECMAScript API	۱۳
۴۹	نگاشت نوع داده	۱-۱۳
۴۹	نگاشت پارامتر تابع	۲-۱۳
۴۹	نگاشت بازگشت تابع	۳-۱۳
۴۹	نگاشت استثنا تابع	۴-۱۳
۴۹	امضاهای فراخوانی رویه	۵-۱۳
۵۷	خطا/استثنا برگشتی	۶-۱۳
۵۸	پیشوند برچسب انطباق	۷-۱۳
۵۹		کتابنامه

پیش‌گفتار

استاندارد «فناوری اطلاعات - تعامل‌پذیری و انقیادهای مراکز ثبت فراداده (MDR-IB) - قسمت ۳: انقیادهای واسط برنامه‌نویسی کاربردی (API)» که پیش‌نویس آن در کمیسیون فنی مربوط، توسط سازمان ملی استاندارد ایران تهیه و تدوین شده و در دویست و نود و سومین اجلاس کمیته ملی استاندارد رایانه و فرآوری داده مورخ ۱۳۹۲/۸/۲۰ مورد تصویب قرار گرفته است اینک به استناد بند یک ماده ۳ قانون اصلاح قوانین و مقررات موسسه استاندارد و تحقیقات صنعتی ایران مصوب بهمن ماه ۱۳۷۱ به عنوان استاندارد ملی منتشر می‌شود.

برای حفظ همگامی و هماهنگی با تحولات و پیشرفت‌های ملی و جهانی در زمینه صنایع، علوم و خدمات، استانداردهای ملی ایران در مواقع لزوم تجدید نظر خواهند شد و هر گونه پیشنهادی که برای اصلاح و تکمیل این استانداردها ارائه شود، هنگام تجدید نظر در کمیسیون فنی مربوط مورد توجه قرار خواهد گرفت.

منبع و ماخذی که در تهیه این استاندارد مورد استفاده قرار گرفته است به شرح زیر است:

ISO/IEC 20944-3:2013, Information technology - Metadata Registries Interoperability and Bindings (MDR-IB) - part 3: API bindings

فناوری اطلاعات - تعامل پذیری و انقیادهای مراکز ثبت فراداده (MDR-IB)^۱ - قسمت ۳: انقیادهای واسط برنامه نویسی کاربردی (API)^۲

۱ هدف و دامنه کاربرد

مجموعه استانداردهای ISO/IEC 20944 به شرح کنونی، واسطهای برنامه نویسی کاربردی (API)^۳ و پروتکل هایی برای تعامل با مرکز ثبت فراداده استاندارد ISO/IEC 11179 می پردازد. هدف از تدوین این استاندارد، تعیین تمهیداتی است که در سراسر انقیادهای API، برای قسمت های مختلف استانداردهای بین المللی ISO/IEC 20944 مشترک هستند و انقیادهای API شخصی را فراهم می کند.

۲ مراجع الزامی

مدارک الزامی زیر حاوی مقرراتی است که در متن این استاندارد ملی ایران به آن ها ارجاع داده شده است. بدین ترتیب آن مقررات جزئی از این استاندارد ملی ایران محسوب می شود. در صورتی که به مدارکی با ذکر تاریخ انتشار ارجاع داده شده باشد، اصلاحیه ها و تجدیدنظرهای بعدی آن مورد نظر این استاندارد ملی ایران نیست. در مورد مدارکی که بدون ذکر تاریخ انتشار به آن ها ارجاع داده شده است، همواره آخرین تجدیدنظر و اصلاحیه های بعدی آن ها مورد نظر است. استفاده از مراجع زیر برای این استاندارد الزامی است:

- 2-1 ISO/IEC 9899:1999, Programming languages — C
- 2-2 ISO/IEC 11404:2007, Information technology — General-Purpose Datatypes (GPD)
- 2-3 ISO/IEC 13886:1996, Information technology — Language-Independent Procedure Calling (LIPC)
- 2-4 ISO/IEC 16262, Information technology — Programming languages, their environments and system software interfaces — ECMAScript language specification
- 2-5 ISO/IEC 20944-1:2013, Information technology — Metadata Registries Interoperability and Bindings (MDR-IB) — Part 1: Framework, common vocabulary, and common provisions for conformance
- 2-6 IETF RFC 3986, Uniform Resource Identifier (URI): Generic Syntax, January 2005

۳ اصطلاحات و تعاریف

در این استاندارد، اصطلاحات و تعاریف استفاده شده در استاندارد ISO/IEC 20944-1، به کار می رود.

1 - Metadata Registries Interoperability and Bindings
2 - Application Programming Interface
3 - Application Programming Interfaces

۴ موارد استفاده این استاندارد

هدف از این استاندارد، ارائه مجموعه مشترک از خدماتها (تمهیدات API مشترک) و انقیادهای API استاندارد شده است، بطوری که برای دسترسی انبارهای MDR، بتوان برنامه‌های قابل حمل نوشت. این برنامه‌ها قابل حمل هستند و این بدین معناست که برنامه مشابه باید بر روی همه محیط‌های عامل بطور مشابه عمل کند و برنامه مشابه باید بتواند به انباره‌های MDR که مطابق با این استاندارد است، دسترسی داشته باشد.

۵ مدل انتزاعی

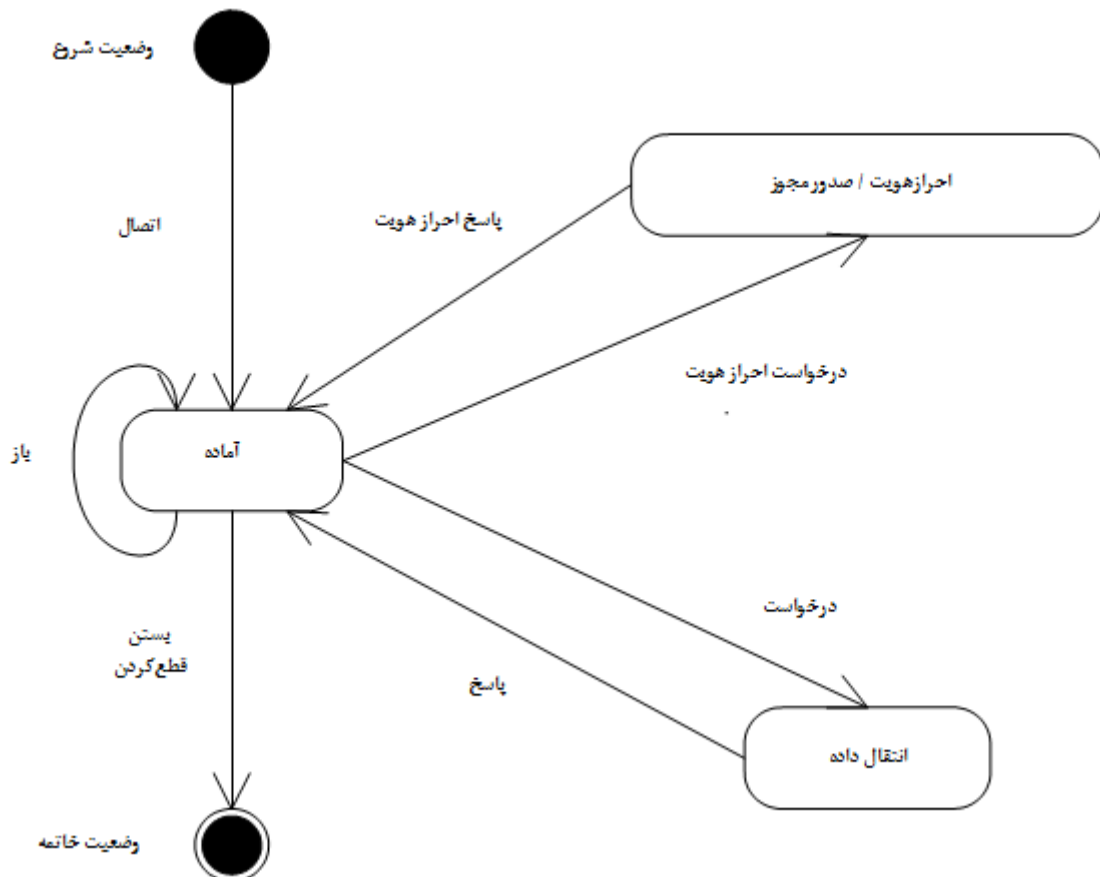
۱-۵ کلیات

انقیادهای API در ادراکشان از خدماتهای دسترسی داده، اشتراک‌هایی دارند. برای مثال، ویژگی‌های مشترک شامل موارد زیر است:

- استفاده از الگوی جلسه برای دسترسی به داده
- استفاده از چارچوب امنیت پارامتری برای پشتیبانی از روش‌های امنیت مختلف
- استفاده از پیمایش سلسله مراتبی برای دسترسی به داده

۲-۵ الگوی جلسه

نمودار وضعیت زیر، وضعیت‌های مختلف خدماتها را توصیف می‌کند:



وضعیت‌ها عبارتند از:

- وضعیت آغازی^۱: وضعیت آغازی نمونه‌ای از API است.
- وضعیت خاتمه^۲: وضعیت خاتمه نمونه‌ای از API است.
- آماده^۳: API برای درخواست‌های خدمات‌ها آماده است.
- احراز هویت - صدورمجوز^۴: API درخواست اصالت‌سنجی - مجوز را پردازش می‌کند.
- انتقال داده^۵: API در خواست انتقال داده را پردازش می‌کند.

رخداده‌ها عبارتند از:

- اتصال^۶: اتصال اولیه را برقرار می‌کند.
- باز^۷: پارامترهای بیشتری را برای ایجاد مدیریت^۸ جدید و محتوای یک مسیر گره جاری برای آن مدیریت اضافه می‌کند.
- بسته^۹: مدیریتی که بوسیله رخداد باز ایجاد شده‌است را از بین می‌برد.
- قطع کردن^{۱۰}: به یک اتصال ضربه می‌زند^{۱۱}.
- درخواست^{۱۲}: یک درخواست انتقال داده است.
- پاسخ^{۱۳}: یک پاسخ انتقال داده است.
- درخواست احراز هویت^{۱۴}: یک درخواست اصالت‌سنجی - صدور مجوز است.
- پاسخ احراز هویت^{۱۵}: یک پاسخ اصالت‌سنجی - صدور مجوز است.

۳-۵ چارچوب امنیت

چارچوب امنیتی یک روش متداول را برای دسترسی به خدمات‌های امنیتی فراهم می‌کند و از یک روش متداول برای پیاده‌سازی خدمات‌های امنیتی پشتیبانی می‌کند. روش دسترسی متداول، از خدمات‌های درخواست-پاسخ اصالت‌سنجی-صدور مجوز (RRAA)^{۱۶} استفاده می‌کند. در خدمات‌های RRAA، یک درخواست برای صدورمجوز، اصالت‌سنجی یا برخی خدمات‌های امنیتی دیگر، قراردادده می‌شود. درخواست، داده را فراهم می‌کند، همانطور که برای خدمات RRAA نیاز است. سپس، خدمات RRAA پاسخی را برای درخواست ارائه می‌دهد. خدمات‌ها در برنامه کاربردی API (مثلاً،

-
- 1 - Initial State
 - 2 - End State
 - 3 - Ready
 - 4 - Authentication - authorization
 - 5 - Data transfer
 - 6 - Connect
 - 7 - Open
 - 8 - Handle
 - 9 - Close
 - 10 - Disconnect
 - 11 - Knocking down
 - 12 - Request
 - 13 - Response
 - 14 - Auth Request
 - 15 - Auth Response
 - 16 - Request-Response Authorization-Authentication

برنامه‌ای که API استفاده می‌کند) و محیط API (مثلاً، خدمات‌ها و زیرساخت‌های اساسی) متقارن هستند و هرکدام ممکن است برای دیگری درخواست ایجاد کند. برای مثال برنامه کاربردی API می‌تواند درخواست‌ها را برای محیط API ایجاد کند و محیط API می‌تواند درخواست‌ها را برای برنامه کاربردی API ایجاد کند. زیرساخت موردپشتیبانی باید یک انتخاب زمان-اجرا و قابل‌پیکربندی بطور پویا فراهم کند بطوری‌که برنامه‌ها نیازی نباشد که برای استفاده از مزایای خدمات‌های RRAA جدید یا روش‌های امنیت جدید، دوباره ترجمه^۱ شوند یا دوباره پیوند برقرار کنند^۲. این استاندارد هیچ الزاماتی را برای مجموعه ویژه از خدمات‌های امنیت ایجاد نمی‌کند. خدمات‌های در دسترس، موقع پیاده‌سازی تعریف می‌شوند^۳.

۴-۵ مسیرهای پیمایش سلسله مراتبی

خدمات‌های API ممکن است از طریق مسیرهای پیمایش سلسله مراتبی به داده دسترسی داشته باشند. مسیرهای پیمایش، همانطور که در استاندارد RFC 3986 تعریف شده است، مبتنی بر شناسه‌های منبع یکنواخت (URI)^۴ هستند. مسیرهای مطلق که در بالای یک انباره شروع می‌شوند در زیربند ۳-۳ از استاندارد RFC 3986 تحت عنوان «مسیرهای مطلق»، مشخص شده‌اند. مسیرهای نسبی که با مسیر جاری در ارتباط هستند، در زیربند ۵ از استاندارد RFC 3986 تحت عنوان «URI‌های نسبی»، مشخص شده‌اند.

۶ خدمات‌ها

۱-۶ استفاده از استاندارد ISO/IEC 13886

نوشتار استاندارد ISO/IEC 13886 صدازدن رویه‌های مستقل از زبان^۵، برای توصیف واسطه‌های خدمت استفاده می‌شود.

۲-۶ خدمات‌های برقراری جلسه^۶

خدمات‌های زیر، جلسه‌ها را با مراکز ثبت فراداده، شروع می‌کنند و خاتمه می‌دهند.

۱-۲-۶ اطلاعات سامانه

خلاصه^۷

```
mdrib_system_info:
procedure
(
    in session: mdrib_handle // session handle
)
returns characterstring,
```

-
- 1 - Recompile
 - 2 - Re-link
 - 3 - Implementation-defined
 - 4 - Uniform Resource Identifier
 - 5 - Language-Independent Procedure Calls
 - 6 - Session
 - 7 - Synopsis

توضیح

خدمات‌های `mdrib_system_info` موارد زیر را بازیابی می‌کند:
خاتمه خط جدید که موقع پیاده‌سازی تعریف می‌شود، زوج‌های نام- مقدار^۱ با توجه به پیکربندی فعلی، محدودیت‌ها و پارامترهای جلسه در `mdrib_handle`. اگر `mdrib_handle` پوچ^۲ باشد، اطلاعات برای همه جلسه‌ها می‌باشد.

پارامترهای زیر بر روی همه پیاده‌سازی‌ها قابل دسترس هستند:
- `apistyle=xxx`، که `xxx` یکی از موارد مقابل می‌باشد: `synchronous` (API همیشه انتظار برای بازگشت را مسدود می‌کند)، `nonblock` (API همیشه فوراً برمی‌گرداند)، `interrupt` (API هنگامی که کامل می‌شود، دچار وقفه می‌شود، `callback` (API هنگامی که کامل می‌شود، به تماس پاسخ می‌دهد)

مثال:

```
// C/C++ illustration
printf(NULL, "mdrib configuration:\n%s", mdrib_system_info());
// sample output
maximum_input_string=4095
supported_types=str8,int,char,long
version_info=3.17
version_description="C implementation with LDAP backend"
```

۲-۲-۶ پیکربندی

خلاصه

```
// values accessible via "._system_info" identifier
data_object_identifier_length // Maximum supported data object identifier length
datatype_family_support // List of families supported
identifier_character_family_support // Allowed data object identifier characters
navigation_identifier_max // Maximum hierarchical navigation identifier length
session_max: Maximum number of simultaneous opened sessions
octet_transfer_max: // Maximum octets of data for a data object
```

```
// error codes accessible via "._error_family" and "._error_list"
error_family // Name of error code system
error_list // List of error codes and meanings
```

توضیح

ویژگی‌های پیکربندی از طریق خدمات‌های `Get` و `Put` از طریق شناسه‌های `._system_info` و `._error_family` و `._error_list` قابل دسترس هستند.

مثال:

```
// C/C++ illustration
if ( strcmp(mdrib_get_value_as_str8(NULL, "._error_family", 0, 0, NULL), "POSIX") == 0 )
{
```

1 - Name-value
2 - Null

```
// Error returns/handling are based upon POSIX standard
// ...
}
```

۶-۲-۳ اتصال

خلاصه

```
mdrib_connect:
procedure
(
    in target: characterstring, // repository to connect to
    in options: characterstring, // connect options
)
returns mdrib_handle,
```

توضیح

یک جلسه جدید را برای انباره داده ایجاد می‌کند که توسط مقصد نام‌گذاری می‌شود و یک رشته نویسه^۱ که موقع پیاده‌سازی تعریف می‌شود، می‌باشد. پارامتر گزینه‌ها، فهرست جداشده با فضای سفید^۲ از زوج‌های نام-مقدار است که مجموعه‌ای از گزینه‌های اتصالات هستند که موقع پیاده‌سازی تعریف می‌شوند. فضای سفید درون رشته نویسه‌های که درون گیومه قراردارند، رهایی می‌یابند، مثلاً، این فضای سفید به‌عنوان جداکننده نام-مقدار، عمل نمی‌کند. اگر موفقیت‌آمیز باشد، یک مدیریت جلسه برای انباره باز می‌گرداند ولی دسترسی را درخواست نمی‌کند (به `mdrib_open` مراجعه کنید). اگر موفقیت‌آمیز نبود، یک مدیریت جلسه پوچ برمی‌گرداند.

مثال:

```
// C/C++ illustration
mdrib_handle sh; // session handle
mdrib_handle ch; // child session handle
sh = mdrib_connect("http://xyz.com/repository_2",
"option_x=\"j k\" option_y=q option_z"); // note: option_x has the value "j k"
if ( sh != NULL )
{
    ch = mdrib_open(sh,"postal_address","read-only");
    // ...
    mdrib_close(ch);
    // ...
    mdrib_disconnect(sh);
}
```

۶-۲-۴ قطع کردن

خلاصه

```
mdrib_disconnect:
procedure
(
    in session: mdrib_handle // session handle
)
returns (state(success,failure)),
```

-
- 1 - Characterstring
 - 2 - Whitespace-separated

توضیح

یک جلسه را انباره داده که در ارتباط با مدیریت جلسه و همه جلسه‌های فرزند است ، قطع می‌کند و می‌بندد. اگر موفقیت‌آمیز باشد، موفقیت و اگر موفقیت‌آمیز نباشد، شکست را برمی‌گرداند.

مثال:

```
// C/C++ illustration
mdrib_handle sh; // session handle
sh = mdrib_connect("http://xyz.com/repository_2",
"option_x=p option_y=q option_z");
// ... open session threads
// ... access metadata
// ... close session threads
mdrib_disconnect(sh);
```

۵-۲-۶ باز

خلاصه

```
mdrib_open:
procedure
(
    in session: mdrib_handle, // session handle
    in node: characterstring, // portion of repository to open
    in options: characterstring, // open options
)
returns (mdrib_handle),
```

توضیح

درون یک جلسه به انباره داده، یک جلسه فرزند را همانطو که اشاره شده، توسط مدیریت جلسه، باز می‌کند. پارامتر گره، نام قسمتی از انباره - یک منظر^۱، است. اگر گره، رشته خالی ("") باشد، سپس جلسه فرزند، نسخه تکراری از جلسه والد است. تقسیم‌بند و نام‌گذاری محتوای انباره‌های در زمان پیاده‌سازی تعریف می‌شود.

پارامتر گزینه‌ها، فهرست جداشده با فضای سفید از زوج‌های نام-مقدار است که مجموعه‌ای از گزینه‌ها از فهرست زیر است:

- "read-only" (or "ro" or "r"): جلسه فرزند با دسترسی فقط خواندنی، باز می‌شود.
- "read-write" (or "rw"): جلسه فرزند با دسترسی خواندن و نوشتن، باز می‌شود.
- "read-seq" (or "rs"): جلسه فرزند با دسترسی ترتیبی خواندن، باز می‌شود. این گزینه ممکن است توسط زیرگزینه "orderint=xxx" که "xxx" ممکن است "breadth"، "depth"، "alphasort" یا "datesort" باشد.
- "write-seq" (or "ws"): جلسه فرزند با دسترسی ترتیبی نوشتن، باز می‌شود.
- "append" (or "a"): جلسه فرزند با دسترسی اضافه کردن نوشتن^۲، باز می‌شود.
- "inherit" (or ""): جلسه فرزند از جلسه والد، خصوصیاتش را به ارث می‌برد.

1 - View

2 - Write-append

اگر `mdrib_open` موفقیت‌آمیز باشد، یک مدیریت به جلسه فرزند و اگر موفقیت‌آمیز نباشد، مدیریت پوچ را برمی‌گرداند.

مثال:

```
// C/C++ illustration
mdrib_handle sh; // session handle
mdrib_handle ch; // child session handle
sh = mdrib_connect("http://xyz.com/repository_2",
"option_x=p option_y=q option_z");
if ( sh != NULL )
{
    ch = mdrib_open(sh,"postal_address","read-only");
    // ...
    mdrib_close(ch);
    // ...
    mdrib_disconnect(sh);
}
```

۶-۲-۶ بسته

خلاصه

```
mdrib_close:
procedure
(
    in mdrib_handle: session, // session handle
)
returns (state(success,failure)),
```

توضیح

یک جلسه که در ارتباط با مدیریت جلسه و همه جلسه‌های فرزند است، را می‌بندد. اگر موفقیت‌آمیز باشد، موفقیت و اگر موفقیت‌آمیز نباشد، شکست را برمی‌گرداند.

مثال:

```
// C/C++ illustration
mdrib_handle sh; // session handle
mdrib_handle ch; // child session handle
sh = mdrib_connect("http://xyz.com/repository_2",
"option_x=p option_y=q option_z");
if ( sh != NULL )
{
    ch = mdrib_open(sh,"postal_address","read-only");
    // ...
    mdrib_close(ch);
    // ...
    mdrib_disconnect(sh);
}
```

۶-۳-۳ خدمات‌های پارامتر جلسه

خدمات‌های زیر ممکن است برای اصلاح و بازیابی پارامترهای جلسه استفاده شوند.

۶-۳-۱ دریافت مسیر

خلاصه

```
mdrib_get_path:
procedure
(
    in session: mdrib_handle, // session handle
)
returns (characterstring),
```

توضیح

مسیر گره جاری را برای جلسه mdrib_handle بازیابی می‌کند. اگر mdrib_get_path موفقیت‌آمیز باشد، رشته‌ای را برمی‌گرداند که شامل مسیر گره جاری است. در غیر این صورت، بازگشت خطا توسط بازگشتی از یک اشاره‌گر پوچ (نه یک رشته خالی) شناسایی می‌شود.

مثال:

```
// C/C++ illustration
mdrib_handle sh; // session handle
mdrib_handle ch; // child session handle
sh = mdrib_connect("http://xyz.com/repository_2",
"option_x=p option_y=q option_z");
if ( sh != NULL )
{
    // open "postal_address"
    ch = mdrib_open(sh,"postal_address","read-only");
    // change path to "city"
    mdrib_put_path(ch,"city");
    // retrieve path, should be "city"
    new_path = mdrib_get_path(ch);
    if ( new_path != "city" )
    {
        // error
    }
}
}
```

۶-۳-۲ قراردادن مسیر

خلاصه

```
mdrib_put_path:
procedure
(
    in session: mdrib_handle, // session handle
    in node: characterstring, // portion of repository
)
returns (state(success,failure)),
```

توضیح

مسیر گره جاری به به مسیر مشخص‌شده توسط جلسه mdrib_handle تغییر می‌دهد. اگر گره، مسیر نسبی باشد، مسیر جدید نسبت به مسیر گره قبلی، نسبی می‌باشد. اگر mdrib_put_path موفقیت‌آمیز باشد، موفقیت را برمی‌گرداند. در غیر این صورت، بازگشت خطا توسط بازگشتی از شکست، برگردانده می‌شود.

مثال:

```
// C/C++ illustration
```

```

mdrib_handle sh; // session handle
mdrib_handle ch; // child session handle
sh = mdrib_connect("http://xyz.com/repository_2",
"option_x=p option_y=q option_z");
if ( sh != NULL )
{
    // open "postal_address"
    ch = mdrib_open(sh,"postal_address","read-only");
    // change path to "city"
    if ( mdrib_put_path(ch,"city") == -1 )
    {
        // error
    }
    // success
}
}

```

۴-۶ خدمت‌های امینت

خدمت‌های زیر پشتیبانی می‌شوند.

۱-۴-۶ درخواست اصالت‌سنجی/صدور مجوز

خلاصه

```

mdrib_request_auth:
procedure
(
    in session: mdrib_handle, // session handle
    in auth_type: characterstring, // auth type
    in auth_options: characterstring, // auth options
)
returns (state(success,failure)),

```

توضیح

از انباره درخواست می‌کند که بوسیله مدیریت جلسه، اعتبار صدورمجوز و/یا اصالت‌سنجی را تامین کند. پارامتر auth_type، فهرست جداشده با فضای سفید از زوج‌های نام- مقدار است که مجموعه‌ای از اعتباراتی که از انباره درخواست می‌شوند را توصیف می‌کند:

- "symmetric": نوع صدورمجوز و/یا اصالت‌سنجی، متقارن است. یعنی هر دو طرف بر روی "word" همانند، مثل کلمه عبور، توافق دارند. پارامتر auth_options فهرستی از کلماتی را مشخص می‌کند که به‌عنوان اعتبارات درخواست دهد.
- "asymmetric": نوع صدورمجوز و/یا اصالت‌سنجی، نامتقارن است. یعنی هر دو طرف بر روی یک مجموعه جدا از "words"، مثل روش کلید عمومی، توافق دارند.
- "challenge": برای رقابت امنیت یک پاسخ را درخواست می‌دهد.
- "identifier": از انباره درخواست می‌کند که فهرستی از شناسه‌ها را برای اصالت‌سنجی عرضه کند.
- "operation": از انباره درخواست می‌کند که فهرستی از عملیات را برای مجوز، عرضه کند.
- "nomad": موافقت و صدورمجوز یک اتصال متحرک^۱ درخواست می‌کند.

پارامتر `auth_options`، فهرست جدانشده با فضای سفید از زوج‌های نام- مقدار است که مجموعه‌ای از گزینه‌های صدورمجوز و/یا اصالت‌سنجی را از فهرست زیر، توصیف می‌کند:

- "Password": کلمه عبور از انباره درخواست می‌شود.
- "key": برای دسترسی نامتقارن، از انباره کلید درخواست می‌شود.
- "identifier": از انباره، شناسه درخواست می‌شود.
- "operation": از انباره، عملیات درخواست می‌شود.
- "nomad_request_id": شناسه مرتبط با درخواست‌دهنده اتصال متحرک، است.
- "nomad_response_id": شناسه مرتبط با پاسخ‌دهنده اتصال متحرک، است.
- "nomad_timeout": مهلت^۱ مرتبط با اتصال متحرک، است.

اگر `mdrib_request_auth` موفقیت‌آمیز باشد، موفقیت را برمی‌گرداند. در غیر این صورت، بازگشت خطا توسط بازگشتی از شکست، شناسایی می‌شود.

مثال:

```
// C/C++ illustration
mdrib_handle sh; // session handle
mdrib_handle ch; // child session handle
sh = mdrib_connect("http://xyz.com/repository_2",
"option_x=p option_y=q option_z");
if ( mdrib_request_auth(sh, "symmetric", "password") !=
"swordfish" )
{
    // ... error, abort because of password mismatch
    return;
}
if ( sh != NULL )
{
    ch = mdrib_open(sh,"postal_address","read-only");
    // ...
    mdrib_close(ch);
    // ...
    mdrib_disconnect(sh);
}
}
```

۲-۴-۶ پاسخ اصالت‌سنجی/ صدورمجوز

خلاصه

```
mdrib_response_auth:
procedure
(
    in session: mdrib_handle, // session handle
    in auth_type: characterstring, // auth type
    auth_handler(): procedure, // auth handler function
)
returns state(success,failure),
```

توضیح

هنگام درخواست برای صدور مجوز و/یا احراز هویت، یک مدیریت برای پاسخهای آنها توسط انباره که بوسیله مدیریت جلسه اشاره شد، ثبت می‌کند.

پارامتر `auth_type`، یک فهرست جداشده با فضای سفید از زوج‌های نام-مقدار است که مجموعه‌ای از اعتبارهایی را توصیف می‌کند که از انباره درخواست شده‌اند. برای فهرست پارامترهای `auth_type`، به زیربند ۶-۴-۱ مراجعه کنید.

پارامتر `auth_type`، یک اشاره‌گر به یک خدمت مدیریت می‌باشد. یک خدمت مدیریت، برای درخواست هر صدور مجوز و/یا احراز هویت که نوع `auth_type` را تطابق می‌دهد، فراخوانی می‌شود. خدمت `auth_handler` با حداقل سه پارامتر فراخوانی می‌شود: مدیریت جلسه، مقدار واقعی `auth_type`، `auth_option` و صفر یا تعداد بیشتری پارامتر.

اگر `mdrib_response_auth` موفقیت‌آمیز باشد، موفقیت را برمی‌گرداند. در غیر این صورت، بازگشت خطا توسط بازگشتی از شکست، شناسایی می‌شود.

مثال:

```
// C/C++ illustration
mdrib_handle sh; // session handle
mdrib_handle ch; // child session handle
sh = mdrib_connect("http://xyz.com/repository_2",
"option_x=p option_y=q option_z");
mdrib_response_auth(sh,"symmetric",handler);
if ( sh != NULL )
{
    ch = mdrib_open(sh,"postal_address","read-only");
    // ...
    mdrib_close(ch);
    // ...
    mdrib_disconnect(sh);
}
// auth handler: returns a password when requested
characterstring handler
(
    mdrib_handle sh,
    characterstring auth_type,
    characterstring auth_options,
    va_list other_parameters
)
{
    if ( auth_type == "symmetric" &&
        auth_options == "password" )
    {
        return "swordfish";
    }
}
```

۵-۶ خدمات‌های انتقال داده

انتقال داده زیر پشتیبانی می‌شود.

```

mdrib_get_value:
procedure
(
  in session: mdrib_handle, // session handle
  in octet_offset: integer, // starting offset for transfer
  in octet_length: integer, // maximum number of octets to transfer
  out octet_transferred: integer, // maximum number of octets transferred
  in src_identifier: characterstring, // src object name
  in dst_object_type: characterstring, // saved value: typeof
  out dst_object_ptr: pointer, // saved value: ptr to
)
returns (state(success,failure)),

```

توضیح

مقداری را می‌گیرد که به یک نوع داده خاص (توسط src_identifier شناسایی می‌شود)، تبدیل می‌شود. اگر مقدار octet_offset، ۱- باشد، اولین هشتایی‌های octet_offset، در انتقال داده نادیده گرفته می‌شود.^۱ اگر octet_length، ۱- نباشد، حداکثر هشتایی‌های octet_length منتقل می‌شوند. تعداد کل هشتایی‌های منتقل شده در octets_transferred برگردانده می‌شود. dst_object_type، نوع داده از شی داده است و جایی است که داده منتقل می‌شود و dst_object_ptr به مقصد داده اشاره می‌کند.

مثال:

```

// C/C++ illustration
mdrib_handle sh; // session handle
mdrib_handle ch; // child session handle
mdrib_type_type value_type; // datatype information
characterstring city_name;
int octet_transferred;
value_type = "characterstring"; // target datatype
sh = mdrib_connect("http://xyz.com/repository_2",
"option_x=p option_y=q option_z");
if ( sh != NULL )
{
  // open "postal_address"
  ch = mdrib_open(sh,"postal_address","read-only");
  success = mdrib_get_value(ch, -1, -1, octet_transferred,
"city", value_type, city_name
);
}

```

۶-۵-۲ Typed get value

```

mdrib_get_value_as_str8:
procedure

```

```

(
    session: mdrib_handle, // session handle
    src_identifier characterstring, // src object name
    in_octet_offset: integer, // starting offset for transfer
    in_octet_length: integer, // maximum number of octets to transfer
    out_octet_transferred: integer, // maximum number of octets transferred
)
returns (str8)
raises (bad_conversion),
mdrib_get_value_as_str16:
procedure
(
    session: mdrib_handle, // session handle
    src_identifier characterstring, // src object name
    in_octet_offset: integer, // starting offset for transfer
    in_octet_length: integer, // maximum number of octets to transfer
    out_octet_transferred: integer, // maximum number of octets transferred
)
returns (str16)
raises (bad_conversion),
mdrib_get_value_as_str32:
procedure session: mdrib_handle, // session handle
src_identifier characterstring, // src object name
in_octet_offset: integer, // starting offset for transfer
in_octet_length: integer, // maximum number of octets to transfer
out_octet_transferred: integer, // maximum number of octets transferred
)
returns (str32)
raises (bad_conversion),
mdrib_get_value_as_int8:
procedure
(
    session: mdrib_handle, // session handle
    src_identifier characterstring, // src object name
)
returns (int8)
raises (bad_conversion),
mdrib_get_value_as_uint8:
procedure
(
    session: mdrib_handle, // session handle
    src_identifier characterstring, // src object name
)
returns (uint8)
raises (bad_conversion),
mdrib_get_value_as_int16:
procedure
(
    session: mdrib_handle, // session handle
    src_identifier characterstring, // src object name

```

```

)
returns (int16)
raises (bad_conversion),
mdrib_get_value_as_uint16:
procedure
(
    session: mdrib_handle, // session handle
    src_identifier characterstring, // src object name
)
returns (uint16)
raises (bad_conversion),
mdrib_get_value_as_int32:
procedure
(
    session: mdrib_handle, // session handle
    src_identifier characterstring, // src object name
)
returns (int32)
raises (bad_conversion),
mdrib_get_value_as_uint32:
procedure
(
    session: mdrib_handle, // session handle
    src_identifier characterstring, // src object name
) returns (uint32)
raises (bad_conversion),
mdrib_get_value_as_int64:
procedure
(
    session: mdrib_handle, // session handle
    src_identifier characterstring, // src object name
)
returns (int64)
raises (bad_conversion),
mdrib_get_value_as_uint64:
procedure
(
    session: mdrib_handle, // session handle
    src_identifier characterstring, // src object name
)
returns (uint64)
raises (bad_conversion),
mdrib_get_value_as_int128:
procedure
(
    session: mdrib_handle, // session handle
    src_identifier characterstring, // src object name
)
returns (int128)
raises (bad_conversion),

```

```

mdrib_get_value_as_uint128:
procedure
(
    session: mdrib_handle, // session handle
    src_identifier characterstring, // src object name
)
returns (uint128)
raises (bad_conversion),
mdrib_get_value_as_real32:
procedure
(
    session: mdrib_handle, // session handle
    src_identifier characterstring, // src object name
)
returns (real32)
raises (bad_conversion),
mdrib_get_value_as_real64:
procedure
(
    session: mdrib_handle, // session handle
    src_identifier characterstring, // src object name
)
returns (real64)
raises (bad_conversion),
mdrib_get_value_as_real80:
procedure The encoding of 16-bit characters is implementation-defined.
6 The encoding of 32-bit characters is implementation-defined.
(
    session: mdrib_handle, // session handle
    src_identifier characterstring, // src object name
)
returns (real80)
raises (bad_conversion),
mdrib_get_value_as_datetime:
procedure
(
    session: mdrib_handle, // session handle
    src_identifier characterstring, // src object name
)
returns (datetime)
raises (bad_conversion),
mdrib_get_value_as_duration:
procedure
(
    session: mdrib_handle, // session handle
    src_identifier characterstring, // src object name
)
returns (duration)
raises (bad_conversion),
mdrib_get_value_as_uri:

```

```

procedure
(
    session: mdrib_handle, // session handle
    src_identifier characterstring, // src object name
)
returns (uri)
raises (bad_conversion),

```

توضیح

مقداری را می‌گیرد که به یک نوع داده خاص (توسط src_identifier شناسایی می‌شود)، تبدیل می‌شود. انواع داده پایین پشتیبانی می‌شوند.

- str8: یک رشته نویسه‌ای^۱ از نویسه‌های ۸ بیتی. برای مثال، نویسه‌هایی با فهرست آماده^۲ از استانداردهای ISO/IEC 8859-1.
- str16: یک رشته نویسه‌ای^۳ از نویسه‌های ۱۶ بیتی. برای مثال، نویسه‌هایی با فهرست آماده^۴ از استانداردهای ISO/IEC 10646 صفحه چندزبانه پایه (BMP)^۵.
- str32: یک رشته نویسه‌ای^۶ از نویسه‌های ۳۲ بیتی. برای مثال، نویسه‌هایی با فهرست آماده^۷ از استانداردهای ISO/IEC 10646.
- int8: عدد صحیح مکمل دو علامت‌دار ۸ بیتی که توسط IEEE 1596.5 تعریف شده است.
- uint8: عدد صحیح مکمل دو بدون علامت ۸ بیتی که توسط IEEE 1596.5 تعریف شده است.
- int16: عدد صحیح مکمل دو علامت‌دار ۱۶ بیتی که توسط IEEE 1596.5 تعریف شده است.
- uint16: عدد صحیح مکمل دو بدون علامت ۱۶ بیتی که توسط IEEE 1596.5 تعریف شده است.
- int32: عدد صحیح مکمل دو علامت‌دار ۳۲ بیتی که توسط IEEE 1596.5 تعریف شده است.
- uint32: عدد صحیح مکمل دو بدون علامت ۳۲ بیتی که توسط IEEE 1596.5 تعریف شده است.
- int64: عدد صحیح مکمل دو علامت‌دار ۶۴ بیتی که توسط IEEE 1596.5 تعریف شده است.
- uint64: عدد صحیح مکمل دو بدون علامت ۶۴ بیتی که توسط IEEE 1596.5 تعریف شده است.
- int128: عدد صحیح مکمل دو علامت‌دار ۱۲۸ بیتی که توسط IEEE 1596.5 تعریف شده است.
- uint128: عدد صحیح مکمل دو بدون علامت ۱۲۸ بیتی که توسط IEEE 1596.5 تعریف شده است.
- real32: یک IEC 60559، عدد ۳۲ بیتی با ممیز شناور.
- real64: یک IEC 60559، عدد ۶۴ بیتی با ممیز شناور.
- real80: یک IEC 60559، عدد ۸۰ بیتی با ممیز شناور.
- datetime: یک مقدار تاریخ و زمان.

1 - Characterstring
 2 - Repertoire
 3 - Characterstring
 4 - Repertoire
 5 - Basic Multilingual Plane
 6 - Characterstring
 7 - Repertoire

- duration: یک مقدار مدت زمان.
- uri: یک شناسه منبع یکنواخت.

اگر `mdrib_get_value_*` موفقیت آمیز باشد، مقداری که مرتبط با `src_identifier` است را بر می‌گرداند. در غیر این صورت، بازگشت خطا توسط اشاره‌گرهای پوچ نشان داده می‌شود که ۱- برای اعداد صحیح علامت‌دار، ۰ برای اعداد صحیح بدون علامت و NaN^۱ برای اعداد حقیقی.

مثال:

```
// C/C++ illustration
mdrib_handle sh; // session handle
mdrib_handle ch; // child session handle
characterstring city_name; // city name in postal address
sh = mdrib_connect("http://xyz.com/repository_2",
"option_x=p option_y=q option_z");
if ( sh != NULL )
{
    // open "postal_address"
    ch = mdrib_open(sh,"postal_address","read-only");
    city_name = mdrib_get_value_as_str8(ch,"city")
}
}
```

۳-۵-۶ قرارداد مقدار

خلاصه

```
integer mdrib_put_value
(
    in session: mdrib_handle, // session handle
    in src_identifier: characterstring, // src object name
    in octet_offset: integer, // starting offset for transfer at destination
    in octet_length: integer, // maximum number of octets to transfer
    out octet_transferred: integer, // maximum number of octets transferred
    in src_object_type: characterstring, // source value: typeof
    in src_object_ptr: pointer, // source value: ptr to
)
```

توضیح

مقداری را می‌گیرد که به یک نوع داده خاص (توسط `src_identifier` شناسایی می‌شود)، تبدیل می‌شود. اگر مقدار `octet_offset`، ۱- باشد، اولین هشتایی‌های `octet_offset`، در انتقال داده در مقصد نادیده گرفته می‌شود^۲. اگر `octet_length`، ۱- نباشد، حداکثر هشتایی‌های `octet_length` منتقل می‌شوند. تعداد کل هشتایی‌های منتقل شده در `octets_transferred` برگردانده می‌شود. `src_object_type`، نوع داده از شی داده است و جایی است که داده از آنجا منتقل می‌شود و `src_object_ptr` به مبدا داده اشاره می‌کند.

مثال:

```
// C/C++ illustration
```

-
- 1 - Not a Number
 - 2 - Skip


```

mdrib_handle sh; // session handle
mdrib_handle ch; // child session handle
mdrib_type_type value_type; // datatype information
characterstring city_name = "New York";
value_type = "characterstring"; // target datatype
sh = mdrib_connect("http://xyz.com/repository_2",
"option_x=p option_y=q option_z");
if ( sh != NULL ) {
    // open "postal_address"
    ch = mdrib_open(sh,"postal_address","read-write");
    status = mdrib_put_value(ch,"city",-1,-1,octet_transferred,
"string", city_name
);
}

```

۴-۵-۶ پاسخ اصالت‌سنجی/ صدور مجوز

خلاصه

```

mdrib_put_value_as_str8:
procedure
(
    session: mdrib_handle, // session handle
    in octet_offset: integer, // starting offset for transfer at destination
    in octet_length: integer, // maximum number of octets to transfer
    out octet_transferred: integer, // maximum number of octets transferred
    dst_identifier: characterstring, // dst object name
    src_value: str8, // src value
)
returns (str8)
raises (bad_conversion),
mdrib_put_value_as_str16:
procedure
(
    session: mdrib_handle, // session handle
    in octet_offset: integer, // starting offset for transfer at destination
    in octet_length: integer, // maximum number of octets to transfer
    out octet_transferred: integer, // maximum number of octets transferred
    dst_identifier: characterstring, // dst object name
    src_value: str16, // src value
)
returns (str16)
raises (bad_conversion),
mdrib_put_value_as_str32:
procedure
(
    session: mdrib_handle, // session handle
    in octet_offset: integer, // starting offset for transfer at destination
    in octet_length: integer, // maximum number of octets to transfer
    out octet_transferred: integer, // maximum number of octets transferred
    dst_identifier: characterstring, // dst object name
    src_value: str32, // src value
)

```

```

returns (str32)
raises (bad_conversion),
mdrib_put_value_as_int8:
procedure
(
    session: mdrib_handle, // session handle
    dst_identifier: characterstring, // dst object name
    src_value: int8, // src value
)
returns (int8)
raises (bad_conversion)mdrib_put_value_as_uint8:
procedure
(
    session: mdrib_handle, // session handle
    dst_identifier: characterstring, // dst object name
    src_value: uint8, // src value
)
returns (uint8)
raises (bad_conversion),
mdrib_put_value_as_int16:
procedure
(
    session: mdrib_handle, // session handle
    dst_identifier: characterstring, // dst object name
    src_value: int16, // src value
)
returns (int16)
raises (bad_conversion),
mdrib_put_value_as_uint16:
procedure
(
    session: mdrib_handle, // session handle
    dst_identifier: characterstring, // dst object name
    src_value: uint16, // src value
)
returns (uint16)
raises (bad_conversion),
mdrib_put_value_as_int32:
procedure
(
    session: mdrib_handle, // session handle
    dst_identifier: characterstring, // dst object name
    src_value: int32, // src value
)
returns (int32)
raises (bad_conversion),
mdrib_put_value_as_uint32:
procedure
(
    session: mdrib_handle, // session handle

```

```

    dst_identifier: characterstring, // dst object name
    src_value: uint32, // src value
)
returns (uint32)
raises (bad_conversion),
mdrib_put_value_as_int64:
procedure
(
    session: mdrib_handle, // session handle
    dst_identifier: characterstring, // dst object name
    src_value: int64, // src value
)
returns (int64)
raises (bad_conversion), mdrib_put_value_as_uint64:
procedure
(
    session: mdrib_handle, // session handle
    dst_identifier: characterstring, // dst object name
    src_value: uint64 // src value
)
returns (uint64)
raises (bad_conversion),
mdrib_put_value_as_int128:
procedure
(
    session: mdrib_handle, // session handle
    dst_identifier: characterstring, // dst object name
    src_value: int128, // src value
)
returns (int128)
raises (bad_conversion),
mdrib_put_value_as_uint128:
procedure
(
    session: mdrib_handle, // session handle
    dst_identifier: characterstring, // dst object name
    src_value: uint128, // src value
)
returns (uint128)
raises (bad_conversion),
mdrib_put_value_as_real32:
procedure
(
    session: mdrib_handle, // session handle
    dst_identifier: characterstring, // dst object name
    src_value: real32, // src value
)
returns (real32)
raises (bad_conversion),
mdrib_put_value_as_real64:

```

```

procedure
(
    session: mdrib_handle, // session handle
    dst_identifier: characterstring, // dst object name
    src_value: real64, // src value
)
returns (real64)
raises (bad_conversion),
mdrib_put_value_as_real80:
procedure
(
    session: mdrib_handle, // session handle
    dst_identifier: characterstring, // dst object name
    src_value: real80, // src value
)
returns (real80)
raises (bad_conversion), mdrib_put_value_as_datetime:
procedure
(
    session: mdrib_handle, // session handle
    dst_identifier: characterstring, // dst object name
    src_value: datetime, // src value
)
returns (datetime)
raises (bad_conversion),
mdrib_put_value_as_duration:
procedure
(
    session: mdrib_handle, // session handle
    dst_identifier: characterstring, // dst object name
    src_value: duration, // src value
)
returns (duration)
raises (bad_conversion),
mdrib_put_value_as_uri:
procedure
(
    session: mdrib_handle, // session handle
    dst_identifier: characterstring, // dst object name
    src_value: uri, // src value
)
returns (uri)
raises (bad_conversion),

```

توضیح

مقداری را می‌گیرد که به یک نوع داده خاص (توسط src_identifier شناسایی می‌شود)، تبدیل می‌شود. انواع داده پایین پشتیبانی می‌شوند.

- str8: یک رشته نویسه‌ای از نویسه‌های ۸ بیتی. برای مثال، نویسه‌هایی با فهرست آماده از استانداردهای ISO/IEC 8859-1.
- str16: یک رشته نویسه‌ای از نویسه‌های ۱۶ بیتی. برای مثال، نویسه‌هایی با فهرست آماده از استانداردهای ISO/IEC 10646 صفحه چندزبانه پایه (BMP) ۱.
- str32: یک رشته نویسه‌ای از نویسه‌های ۳۲ بیتی. برای مثال، نویسه‌هایی با فهرست آماده از استانداردهای ISO/IEC 10646.
- int8: عدد صحیح مکمل دو علامت‌دار ۸ بیتی که توسط IEEE 1596.5 تعریف شده است.
- uint8: عدد صحیح مکمل دو بدون علامت ۸ بیتی که توسط IEEE 1596.5 تعریف شده است.
- int16: عدد صحیح مکمل دو علامت‌دار ۱۶ بیتی که توسط IEEE 1596.5 تعریف شده است.
- uint16: عدد صحیح مکمل دو بدون علامت ۱۶ بیتی که توسط IEEE 1596.5 تعریف شده است.
- int32: عدد صحیح مکمل دو علامت‌دار ۳۲ بیتی که توسط IEEE 1596.5 تعریف شده است.
- uint32: عدد صحیح مکمل دو بدون علامت ۳۲ بیتی که توسط IEEE 1596.5 تعریف شده است.
- int64: عدد صحیح مکمل دو علامت‌دار ۶۴ بیتی که توسط IEEE 1596.5 تعریف شده است.
- uint64: عدد صحیح مکمل دو بدون علامت ۶۴ بیتی که توسط IEEE 1596.5 تعریف شده است.
- int128: عدد صحیح مکمل دو علامت‌دار ۱۲۸ بیتی که توسط IEEE 1596.5 تعریف شده است.
- uint128: عدد صحیح مکمل دو بدون علامت ۱۲۸ بیتی که توسط IEEE 1596.5 تعریف شده است.
- real32: یک IEC 60559، عدد ۳۲ بیتی با ممیز شناور.
- real64: یک IEC 60559، عدد ۶۴ بیتی با ممیز شناور.
- real80: یک IEC 60559، عدد ۸۰ بیتی با ممیز شناور.
- datetime: یک مقدار تاریخ و زمان.
- duration: یک مقدار مدت زمان.
- uri: یک شناسه منبع یکنواخت.

اگر *mdrib_put_value_ موفقیت آمیز باشد، مقداری که مرتبط با dst_identifer است را بر می‌گرداند. در غیر این صورت، بازگشت خطا توسط اشاره‌گرهای پوچ نشان داده می‌شود که "۱-" برای اعداد صحیح علامت‌دار، ۰ برای اعداد صحیح بدون علامت و NaN برای اعداد حقیقی.

مثال:

```
// C/C++ illustration
mdrib_handle sh; // session handle
mdrib_handle ch; // child session handle
characterstring city_name; // city name in postal address
sh = mdrib_connect("http://xyz.com/repository_2",
"option_x=p option_y=q option_z");
if ( sh != NULL )
{
    // open "postal_address"
```

```

ch = mdrib_open(sh,"postal_address","read-only");
city_name = mdrib_put_value_as_str8(ch,
"city","New York")
}

```

۶-۶ متفرقه

۱-۶-۶ خدمت ساختن شی^۱

خلاصه

```

mdrib_make_object:
procedure
(
  in session: mdrib_handle, // session handle
  in octet_offset: integer, // starting offset for transfer at destination
  in octet_length: integer, // maximum number of octets to transfer
  out octet_transferred: integer, // maximum number of octets transferred
  in src_object_type: characterstring, // source
  value: typeof, // the type of object
  in src_object_ptr: pointer, // source value: ptr to
)
returns (state(success,failure)),

```

توضیح

یک شی جدید که هنگام پیاده‌سازی تعریف می‌شود، ایجاد می‌کند. پارامترها، همانند mdrib_put_object هستند.

اگر mdrib_make_object موفقیت‌آمیز باشد، موفقیت را برمی‌گرداند. در غیر این صورت، بازگشت خطا توسط بازگشتی از شکست، شناسایی می‌شود.

مثال:

```

// C/C++ illustration
mdrib_handle sh; // session handle
mdrib_handle ch; // child session handle
sh = mdrib_connect("http://xyz.com/repository_2",
"option_x=p option_y=q option_z");
if ( sh != NULL )
{
  // open "postal_address"
  ch = mdrib_open(sh,"postal_address","read-write");
  city_name = mdrib_make_object(ch,"city",-1,-1,NULL,
"city_name", string, city_name);
}

```

۲-۶-۶ خدمت حذف شی^۲

خلاصه

mdrib_delete_object:

-
- 1 - Make Object Service
 - 2 - Delete Object Service

```

procedure
(
    in session: mdrib_handle, // session handle
    in src_identifier: characterstring, // src object name
    in qualifier: characterstring, // delete service qualifier
)
returns (state(success,failure)),

```

توضیح

یک شی را حذف می‌کند. `src_identifier`، شی برای حذف را شناسایی می‌کند. اگر `mdrib_remove_object` موفقیت‌آمیز باشد، موفقیت را برمی‌گرداند. در غیر این صورت، بازگشت خطا توسط بازگشتی از شکست، شناسایی می‌شود.

مثال:

```

// C/C++ illustration
mdrib_handle sh; // session handle
mdrib_handle ch; // child session handle
sh = mdrib_connect("http://xyz.com/repository_2",
"option_x=p option_y=q option_z");
if ( sh != NULL )
{
    // open "postal_address"
    ch = mdrib_open(sh,"postal_address","read-only");
    // make object "city"
    if ( mdrib_remove_object(ch, "city", "") == -1 )
    {
        // error
    }
    // success
}

```

۳-۶-۶ خدمت پیوند شی^۱

خلاصه

```

mdrib_link_object:
procedure
(
    session: mdrib_handle, // session handle
    in src_identifier: characterstring, // src object name
    in dst_identifier: characterstring, // dst object name
    in qualifier: characterstring, // link type: soft, hard
)
returns (state(success,failure)),

```

توضیح

شی موجود را به یک نام جدید پیوند می‌دهد. `src_identifier`، نام شی موجود است. `dst_identifier`، نام شی جدید است. `link_type`، نوع پیوند است که مقدار آن `“hard”` (منابع یکسان) یا `“soft”` (به‌طور خودکار پیوند را ارجاع می‌دهد) است.

اگر `mdrib_link_object` موفقیت‌آمیز باشد، موفقیت را برمی‌گرداند. در غیر این صورت، بازگشت خطا توسط بازگشتی از شکست، شناسایی می‌شود.

مثال:

```
// C/C++ illustration
mdrib_handle sh; // session handle
mdrib_handle ch; // child session handle
sh = mdrib_connect("http://xyz.com/repository_2",
"option_x=p option_y=q option_z");
if ( sh != NULL )
{
    // open "postal_address"
    ch = mdrib_open(sh,"postal_address","read-only");
    // link object "state" to "province"
    if ( mdrib_link_object(ch, "state", "province", "soft") == -1 )
    {
        // error
    }
    // success
}
}
```

۴-۶-۶ خدمت یافتن خصوصیت^۱

خلاصه

```
mdrib_property_find
(
    in session: mdrib_handle, // session handle
    in src_identifier: characterstring, // src object name (wildcard)
    in qualifier: characterstring, // propfind service qualifier
)
returns (characterstring),
```

توضیح

همه‌اشیایی که با `src-identifier` مطابقت دارند، را فهرست می‌کند. برای فهرست پارامترها، به خدمات‌های استاندارد RFC 4518 PROPFIND مراجعه کنید.

اگر `mdrib_list_object` موفقیت‌آمیز باشد، رشته نویسه را برمی‌گرداند. در غیر این صورت، بازگشت خطا توسط اشاره‌گر پوچ، شناسایی می‌شود.

مثال:

```
// C/C++ illustration
mdrib_handle sh; // session handle
mdrib_handle ch; // child session handle
characterstring match_list; // resulting list
sh = mdrib_connect("http://xyz.com/repository_2",
"option_x=p option_y=q option_z");
if ( sh != NULL )
{
    // open "postal_address"
    ch = mdrib_open(sh,"postal_address","read-only");
```



```

// list objects
match_list = mdrib_prop_find(ch, "*", "");
if ( match_list == NULL )
{
// error
}
// success
}

```

۵-۶-۶ خدمت قفل کردن شی^۱

خلاصه

```

mdrib_lock_object
(
  in session: mdrib_handle, // session handle
  in src_identifier: characterstring, // src object name
  in qualifier: characterstring, // lock service qualifier
)
returns (state(success,failure)),

```

توضیح

شی src_identifier را مطابق با مشخصات src_lock قفل می‌کند. برای مشخصات LOCK، به RFC 4518 مراجعه کنید.

مثال:

```

// Note 1: See RFC 4518 for lock specification
// Note 2: This example adapted from RFC 4518, subclause 9.10.7
// Locking the resource "0123/value_domain_datatype" ...
mdrib_lock_object(mdrib_handle,
  "0123/value_domain_datatype",
  "<?xml version='1.0' encoding='utf-8' ?> "
  "<D:lockinfo xmlns:D='DAV:'> "
  " <D:lockscope><D:exclusive/></D:lockscope>"
  " <D:locktype><D:write/></D:locktype>"
  " <D:owner>"
  " <D:href>http://example.org/~ff/contact.html</D:href>"
  " </D:owner>"
  " </D:lockinfo>"
);

```

۶-۶-۶ خدمت گشودن قفل شی^۲

خلاصه

```

mdrib_unlock_object
(
  in session: mdrib_handle, // session handle
  in src_identifier: characterstring, // src object name
  in qualifier: characterstring, // unlock service qualifier
)

```

1 - Lock Object Service
2 - Unlock Object Service

returns (state(success,failure)),

توضیح

شی src_identifier را مطابق با مشخصات src_unlock قفل‌گشایی می‌کند. برای مشخصات UNLOCK، به RFC 4518 مراجعه کنید.

مثال:

```
// Note 1: See RFC 4518 for unlock specification
// Note 2: This example adapted from RFC 4518, subclause 9.11.2
// Locking the resource "0123/value_domain_datatype" ...
mdrib_unlock_object(mdrib_handle,
    "0123/value_domain_datatype",
    "");
);
```

۷-۶-۶ خدمت نسخه‌برداری شی^۱

خلاصه

```
mdrib_copy_object
(
    in session: mdrib_handle, // session handle
    in src_identifier: characterstring, // src object name
    in dst_identifier: characterstring, // dst object name
    in qualifier: characterstring, // copy service qualifier
)
returns (state(success,failure)),
```

توضیح

شی src_identifier را با نام جدید dst_identifier، نسخه‌برداری می‌کند. برای مشخصات COPY، به RFC 4518 مراجعه کنید.

مثال:

```
// Note 1: See RFC 4518 for copy specification
// Note 2: This example adapted from RFC 4518, subclause 9.8.7
// Copying resource "0123/value_domain_datatype" to "4567/value_domain_datatype"
// with no overwrite
mdrib_copy_object(mdrib_handle,
    "0123/value_domain_datatype",
    "4567/value_domain_datatype",
    "Overwrite: F",
);
```

۸-۶-۶ خدمت حرکت شی^۲

خلاصه

1 - Copy Object Service
2 - Move Object Service

```

mdrib_move_object
(
    in session: mdrib_handle, // session handle
    in src_identifier: characterstring, // src object name
    in dst_identifier: characterstring, // dst object name
    in qualifier: characterstring, // move service qualifier
)
returns (state(success,failure)),

```

توضیح

شی `src_identifier` را با نام جدید `dst_identifier`، تغییر نام می‌دهد. برای مشخصات MOVE، به RFC 4518 مراجعه کنید.

مثال:

```

// Note 1: See RFC 4518 for move specification
// Note 2: This example adapted from RFC 4518, subclause 9.9.5
// Moving resource "0123/value_domain_datatype" to "4567/value_domain_datatype",
// i.e., renaming it
mdrib_move_object(mdrib_handle,
    "0123/value_domain_datatype",
    "4567/value_domain_datatype",
    "",
);

```

۷ انقیادها

این استاندارد مدل مفهومی مشترک و خدمات‌های داده مشترک را سرتاسر همه انقیادهای کدگذاری توصیف می‌کند. یک انقیاد API انطباقی باید بر الزامات توصیف شده در بندهای ۴، ۵، ۶، ۸ و ۹، منطبق باشد.

۸ اداره^۱

هیچ الزام مدیریتی^۲ وجود ندارد.

۹ انطباق

۹-۱ نمونه انطباق API

نمونه انطباق برای مجموعه استانداردهای ISO/IEC 20944 شامل قواعد انطباق زیر است: برنامه کاربردی API، محیط API

تعاریف: پشتیبانی^۳، استفاده^۴، آزمون^۵، دسترسی^۶، تفحص^۷

-
- 1 - Administration
 - 2 - Administrative
 - 3 - Support
 - 4 - Use
 - 5 - Test
 - 6 - Access
 - 7 - Probe

اصطلاحات زیر در محتوای انطباق API برای شرکت‌کنندگان در تبادل داده، تعریف می‌شود:
ویژگی “supported” یکی از آنهایی است که توسط محیط API پیاده‌سازی شده و ممکن است توسط هرکدام از برنامه‌های کاربردی API، استفاده شود.

- یک ویژگی “used” می‌باشد اگر توسط برنامه کاربردی API، خوانده یا نوشته شود و یا برویش عملی انجام شود.
- یک ویژگی “tested” می‌باشد اگر برنامه کاربردی API، در مورد وجود آن ویژگی در محیط API، تحقیق کند.
- یک ویژگی “accessed” می‌باشد اگر برنامه کاربردی API، قصد خواند یا نوشتن داده مرتبط با آن ویژگی را داشته باشد.
- یک ویژگی “probed” می‌باشد اگر برنامه کاربردی API، بطور ضمنی وجود آن ویژگی را بوسیله تلاش برای استفاده از ویژگی (به اصطلاح “use” در بالا مراجعه شود) درون یک محیط API که بازیابی خطا را اجازه می‌دهد، بیازماید.

یادآوری - انطباق API، الزاماتی را بر روی همه شرکت‌کنندگان تبادل داده ایجاد می‌کند: محیط API (پیاده‌سازی واسط، خدمت‌ها، منابع و غیره از انقیادهای API) و برنامه کاربردی API (برنامه‌های کاربردی که از انقیادهای API استفاده می‌کنند).

۲-۹ برنامه کاربردی API

یک برنامه کاربردی که بسیار منطبق بر API است، باید ویژگی‌های API که برطبق الزامات انقیادهای API است، استفاده کند. یک برنامه کاربردی که بسیار منطبق بر API است، نباید رفتاری داشته باشد که در هنگام پیاده‌سازی تعریف شود.

یک برنامه کاربردی منطبق بر API باید ویژگی‌های API که برطبق الزامات انقیادهای API است، استفاده کند.

۳-۹ محیط API

یک محیط که بسیار منطبق بر API است، باید همه ویژگی‌های API که برطبق الزامات انقیادهای API است، پیاده‌سازی کند. یک محیط که بسیار منطبق بر API است، نباید توسعه‌های^۱ انقیادهای API را پیاده‌سازی کند.

یک محیط منطبق بر API، باید همه ویژگی‌های API که برطبق الزامات انقیادهای API است، پیاده‌سازی کند. یک محیط منطبق بر API، ممکن است توسعه‌های انقیادهای API را پیاده‌سازی کند. یک محیط منطبق بر API، ممکن است خدماتی را برای آزمون، دسترسی و/یا تفحص برای قابلیت‌های اضافی، فراهم کند.

۴-۹ برجسب‌های انطباق

برجسب‌های زیر انواع خاصی از انطباق با این استاندارد را نشان می‌دهد:

- الگوی ۱: "ISO/IEC 20944-3/B/prefix" که "prefix" نوعی از انقیاد است که در بندهای ۱۱ و بعد از آن تعریف می‌شود. مثلاً "ISO/IEC 20944-3/B/C"، مربوط به انقیاد C API است.
 - الگوی ۲: "ISO/IEC 20944-3/B/prefix/role" که "prefix" نوعی از انقیاد است که در بندهای ۱۱ و بعد از آن تعریف می‌شود و "role" یکی از "A" یا "E" مربوط به برنامه کاربردی API و محیط API است که مطابق با تعریف همین بند است. مثلاً "ISO/IEC 20944-3/B/C/A"، مربوط به برنامه کاربردی با استفاده از انقیاد C API است.
- محل قراردادن برچسب‌های انطباق خارج از دامنه کاربرد این استاندارد است.

۱۰ محفوظ برای استاندارد سازی آینده

این بند برای استانداردسازی آینده خالی نگه داشته شده است.

۱۱ انقیاد C API

برای استانداردسازی API مستقل از انقیاد، به زبان برنامه‌نویسی C، مطابق با قواعد زیر، نگاشت می‌شود. انواع داده^۱ انقیاد C API منطبق بر زبان برنامه‌نویسی C است.

۱-۱۱ نگاشت نوع داده

انواع داده‌ی API مستقل از انقیاد همانند پایین نگاشت می‌شوند:

انقیاد C API	مستقل از انقیاد
mdrib_handle	mdrib_handle
string (maximum length 32767)	characterstring
void *	pointer
unsigned char []	str8
wchar_t[]	str16
wchar_t[]	str32
int8_t, uint8_t, int16_t, uint16_t, int32_t, uint32_t, int64_t, uint64_t, int128_t, uint128_t	int8, uint8, int16, uint16, int32, uint32, int64, uint64, int128, uint128
time_t	datetime
time_t	duration
unsigned char []	uri
float, double, long double	real32, real64, real80
(void *)()	procedure

۲-۱۱ نگاشت پارامتر تابع

پارامترهای API مستقل از انقیاد، مشابه پارامترها برای انقیاد C است.

۳-۱۱ نگاشت برگشت تابع

مقادیر برگشتی API مستقل از انقیاد، مانند زیر نگاشت می‌شوند:

انقیاد C API	مستقل از انقیاد
int, success=0, failure=-1	state(success,failure)
string (maximum length 65535)	characterstring
unsigned char []	str8
wchar_t []	str16
wchar_t []	str32
int8_t, uint8_t, int16_t, uint16_t, int32_t, uint32_t, int64_t, uint64_t, int128_t, uint128_t	int8, uint8, int16, uint16, int32, uint32, int64, uint64, int128, uint128
time_t	datetime
time_t	duration
unsigned char []	uri
float, double, long double	real32, real64, real80

۴-۱۱ نگاشت استثنا تابع

استثناهای تابع، مانند زیر نگاشت می‌شوند:

انقیاد C API	مستقل از انقیاد
errno is set	conversion

۵-۱۱ امضای فراخوانی رویه^۱

۱-۵-۱۱ خدمت‌های استقرار جلسه

۱-۱-۵-۱۱ اطلاعات سامانه

خلاصه

```
unsigned char *mdrib_system_info
(
    mdrib_handle session // session handle
)
```

۲-۱-۵-۱۱ پیکربندی

خلاصه

```
int mdrib_configure
(
    mdrib_handle session, // session handle
    unsigned char *parameter_name, // set parameter "name"
    unsigned char *parameter_value // to the value "value"
)
```

۳-۱-۵-۱۱ اتصال

خلاصه

```
mdrib_handle mdrib_connect
(
    unsigned char *target, // repository to connect to
```

```
    unsigned char *options // connect options
)

```

۲-۵-۱۱ قطع

خلاصه

```
int mdrib_disconnect
(
    mdrib_handle session // session handle
)

```

۱-۲-۵-۱۱ باز

خلاصه

```
mdrib_handle mdrib_open
(
    mdrib_handle session, // session handle
    unsigned char *node, // portion of repository to open
    unsigned char *options // connect options
)

```

۲-۲-۵-۱۱ بسته

خلاصه

```
int mdrib_close
(
    mdrib_handle session // session handle
)

```

۳-۵-۱۱ خدمات‌های پارامتر جلسه

۱-۳-۵-۱۱ دریافت مسیر

خلاصه

```
unsigned char *mdrib_get_path
(
    mdrib_handle session // session handle
)

```

۴-۵-۱۱ خدمات‌های امنیت

۱-۴-۵-۱۱ درخواست احراز هویت / صدور مجوز

خلاصه

```
int mdrib_request_auth
(
    mdrib_handle session, // session handle
    unsigned char *auth_type, // auth type
    unsigned char *auth_options // auth options
)

```

۲-۴-۵-۱۱ پاسخ احراز هویت / صدور مجوز

خلاصه

```

int mdrib_response_auth
(
    mdrib_handle session, // session handle
    unsigned char *auth_type, // auth type
    (*int)() auth_handler() // auth handler function
)

```

۵-۵-۱۱ خدمات‌های انتقال داده

۱-۵-۵-۱۱ دریافت مقدار

خلاصه

```

int mdrib_get_value
(
    mdrib_handle session, // [in] session handle
    unsigned char *src_identifier, // [in] src object name
    unsigned char *dst_label_type, // [in] saved label: typeof
    mdrib_object *dst_label_ptr, // [out] saved label: ptr to
    unsigned char *dst_type_type, // [in] saved datatype: typeof
    mdrib_object *dst_type_ptr, // [out] saved datatype: ptr to
    unsigned char *dst_object_type, // [in] saved value: typeof
    mdrib_object *dst_object_ptr, // [out] saved value: ptr to
    unsigned char *dst_proplist_type, // [in] saved proplist: typeof
    mdrib_object *dst_proplist_ptr, // [out] saved proplist: ptr to
)

```

Typed get value ۲-۵-۵-۱۱

خلاصه

```

unsigned char *mdrib_get_value_as_str8
(
    mdrib_handle session, // [in] session handle
    unsigned char *src_identifier // [in] src object name
)
wchar_t *mdrib_get_value_as_str16
(
    mdrib_handle session, // [in] session handle
    unsigned char *src_identifier // [in] src object name
)
wchar_t *mdrib_get_value_as_str32
(
    mdrib_handle session, // [in] session handle
    unsigned char *src_identifier // [in] src object name
)
int8_t mdrib_get_value_as_int8
(
    mdrib_handle session, // [in] session handle
    unsigned char *src_identifier // [in] src object name
)
uint8_t mdrib_get_value_as_uint8
(

```



```

    mdrib_handle session, // [in] session handle
    unsigned char *src_identifier // [in] src object name
)
int16_t mdrib_get_value_as_int16
(
    mdrib_handle session, // [in] session handle
    unsigned char *src_identifier // [in] src object name
)
uint16_t mdrib_get_value_as_uint16
(
    mdrib_handle session, // [in] session handle
    unsigned char *src_identifier // [in] src object name
)
int32_t mdrib_get_value_as_int32
(
    mdrib_handle session, // [in] session handle
    unsigned char *src_identifier // [in] src object name
)
uint32_t mdrib_get_value_as_uint32
(
    mdrib_handle session, // [in] session handle
    unsigned char *src_identifier // [in] src object name
)
int64_t mdrib_get_value_as_int64
(
    mdrib_handle session, // [in] session handle
    unsigned char *src_identifier // [in] src object name
)
uint64_t mdrib_get_value_as_uint64
(
    mdrib_handle session, // [in] session handle
    unsigned char *src_identifier // [in] src object name
)
int128_t mdrib_get_value_as_int128
(
    mdrib_handle session, // [in] session handle
    unsigned char *src_identifier // [in] src object name
)
uint128_t mdrib_get_value_as_uint128
(
    mdrib_handle session, // [in] session handle
    unsigned char *src_identifier // [in] src object name
)
float mdrib_get_value_as_real32
(
    mdrib_handle session, // [in] session handle
    unsigned char *src_identifier // [in] src object name
)
double mdrib_get_value_as_real64
(

```

```

    mdrib_handle session, // [in] session handle
    unsigned char *src_identifier // [in] src object name
)
long double mdrib_get_value_as_real80
(
    mdrib_handle session, // [in] session handle
    unsigned char *src_identifier // [in] src object name
)

```

۳-۵-۵-۱۱ قراردادن مقدار

خلاصه

```

int mdrib_put_value
(
    mdrib_handle session, // [in] session handle
    unsigned char *src_identifier, // [in] src object name
    unsigned char *src_label_type, // [in] assigned label: typeof
    mdrib_object src_label_ptr, // [in] assigned label: ptr to
    unsigned char *src_type_type, // [in] assigned datatype: typeof
    mdrib_object src_type_ptr, // [in] assigned datatype: ptr to
    unsigned char *src_object_type, // [in] assigned value: typeof
    mdrib_object src_object_ptr, // [in] assigned value: ptr to
    unsigned char *src_proplist_type, // [in] assigned proplist: typeof
    mdrib_object src_proplist_ptr, // [in] assigned proplist: ptr to
)

```

Typed put value ۴-۵-۵-۱۱

خلاصه

```

int mdrib_put_value_as_str8
(
    mdrib_handle session, // [in] session handle
    unsigned char *dst_identifier, // [in] dst object name
    str8 src_value // [in] src value
)
int mdrib_put_value_as_str16
(
    mdrib_handle session, // [in] session handle
    unsigned char *dst_identifier, // [in] dst object name
    str16 src_value // [in] src value
)
int mdrib_put_value_as_str32
(
    mdrib_handle session, // [in] session handle
    unsigned char *dst_identifier, // [in] dst object name
    str32 src_value // [in] src value
)
int mdrib_put_value_as_int8
(
    mdrib_handle session, // [in] session handle
    unsigned char *dst_identifier, // [in] dst object name
)

```

```

    int8 src_value // [in] src value
)
int mdrib_put_value_as_uint8
(
    mdrib_handle session, // [in] session handle
    unsigned char *dst_identifier, // [in] dst object name
    uint8 src_value // [in] src value
)
int mdrib_put_value_as_int16
(
    mdrib_handle session, // [in] session handle
    unsigned char *dst_identifier, // [in] dst object name
    int16 src_value // [in] src value
)
int mdrib_put_value_as_uint16
(
    mdrib_handle session, // [in] session handle
    unsigned char *dst_identifier, // [in] dst object name
    uint16 src_value // [in] src value
)
int mdrib_put_value_as_int32
(
    mdrib_handle session, // [in] session handle
    unsigned char *dst_identifier, // [in] dst object name
    int32 src_value // [in] src value
)
int mdrib_put_value_as_uint32
(
    mdrib_handle session, // [in] session handle
    unsigned char *dst_identifier, // [in] dst object name
    uint32 src_value // [in] src value
)
int mdrib_put_value_as_int64
(
    mdrib_handle session, // [in] session handle
    unsigned char *dst_identifier, // [in] dst object name
    int64 src_value // [in] src value
)
int mdrib_put_value_as_uint64
(
    mdrib_handle session, // [in] session handle
    unsigned char *dst_identifier // [in] dst object name
    uint64 src_value // [in] src value
)
int mdrib_put_value_as_int128
(
    mdrib_handle session, // [in] session handle
    unsigned char *dst_identifier, // [in] dst object name
    int128 src_value // [in] src value
)

```

```

int mdrib_put_value_as_uint128
(
    mdrib_handle session, // [in] session handle
    unsigned char *dst_identifier, // [in] dst object name
    uint128_t src_value // [in] src value
)
int mdrib_put_value_as_real32
(
    mdrib_handle session, // [in] session handle
    unsigned char *dst_identifier, // [in] dst object name
    float src_value // [in] src value
)
int mdrib_put_value_as_real64
(
    mdrib_handle session, // [in] session handle
    unsigned char *dst_identifier, // [in] dst object name
    double src_value // [in] src value
)
int double mdrib_put_value_as_real80
(
    mdrib_handle session, // [in] session handle
    unsigned char *dst_identifier, // [in] dst object name
    long double src_value // [in] src value
)

```

۱۱-۵-۶ متفرقه

۱۱-۵-۶-۱ خدمت ساختن شی

خلاصه

```

integer mdrib_make_object
(
    mdrib_handle session, // [in] session handle
    unsigned char *src_identifier, // [in] src object name
    unsigned char *src_label_type, // [in] assigned label: typeof
    mdrib_object src_label_ptr, // [in] assigned label: ptr to
    unsigned char *src_type_type, // [in] assigned datatype: typeof
    mdrib_object src_type_ptr, // [in] assigned datatype: ptr to
    unsigned char *src_object_type, // [in] assigned value: typeof
    mdrib_object src_object_ptr, // [in] assigned value: ptr to
    unsigned char *src_proplist_type, // [in] assigned proplist: typeof
    mdrib_object src_proplist_ptr, // [in] assigned proplist: ptr to
)

```

۱۱-۵-۶-۲ خدمت حذف شی

خلاصه

```

integer mdrib_delete_object
(
    mdrib_handle session, // [in] session handle
    unsigned char *src_identifier, // [in] src object name

```

```
unsigned char *qualifier // [in] delete service qualifier
)
```

۳-۶-۵-۱۱ خدمت پیوند شی

خلاصه

```
integer mdrib_link_object
(
  mdrib_handle session, // [in] session handle
  unsigned char *src_identifier, // [in] src object name
  unsigned char *dst_identifier, // [in] dst object name
  unsigned char *qualifier // [in] link type: soft, hard
)
```

۴-۶-۵-۱۱ خدمت یافتن خصوصیت

خلاصه

```
characterstring mdrib_property_find
(
  mdrib_handle session, // [in] session handle
  unsigned char *src_identifier, // [in] src object name (wildcard)
  unsigned char *qualifier // [in] propfind service qualifier
)
```

۵-۶-۵-۱۱ خدمت قفل کردن شی

خلاصه

```
integer mdrib_lock_object
(
  mdrib_handle session, // [in] session handle
  unsigned char *src_identifier, // [in] src object name
  unsigned char *qualifier // [in] lock service qualifier
)
```

۶-۶-۵-۱۱ خدمت قفل گشودن شی

خلاصه

```
integer mdrib_unlock_object
(
  mdrib_handle session, // [in] session handle
  unsigned char *src_identifier, // [in] src object name
  unsigned char *qualifier // [in] unlock service qualifier
)
```

۷-۶-۵-۱۱ خدمت نسخه برداری شی

خلاصه

```
integer mdrib_copy_object
(
  mdrib_handle session, // [in] session handle
  unsigned char *src_identifier, // [in] src object name
  unsigned char *dst_identifier, // [in] dst object name
  unsigned char *qualifier // [in] copy service qualifier
)
```

integer mdrib_move_object

```
(
    mdrib_handle session, // [in] session handle
    unsigned char *src_identifier, // [in] src object name
    unsigned char *dst_identifier, // [in] dst object name
    unsigned char *qualifier // [in] move service qualifier
)
```

۱۱-۶ خطا/استثنا برگشتی

- خطا/استثناهای بازگشتی زیر از طریق متغیر erro عبور داده می‌شوند:^۱
- MDRIB_ERR_ACCESS: مجوز، عملیات در حال اجرا را نادیده گرفت.
 - MDRIB_ERR_AGAIN: در حالت مشخص شده، منابع فعلاً موجود نیست به دلیل قفل شدن سازوکار، دوباره امتحان کنید.
 - MDRIB_ERR_BADH: مدیریت جلسه یا اتصال، معتبر نیست.
 - MDRIB_ERR_INVALID: مقدار یکی از گزینه های خطای نحوی غیرقانونی دارد یا قواعد نحوی پارامتر مقصد معتبر نیست.
 - MDRIB_ERR_IO: هنگام اجرای عملیات، خطای ورودی/خروجی^۲ رخ داده است.
 - MDRIB_ERR_NOENT: مقصد یا مسیر مشخص شده وجود ندارد.
 - MDRIB_ERR_NOMEM: برای بازکردن اتصال، حافظه کافی وجود ندارد.
 - MDRIB_ERR_NOSPC: برای ایجاد انباره مشخص شده، فضای کافی وجود ندارد.
 - MDRIB_ERR_NOTOBJ: مولفه مسیر به شی معتبر اشاره نمی‌کند.
 - MDRIB_ERR_PERM: عملیات به علت نقض دسترسی مجاز نیست.
 - MDRIB_ERR_NOSUP: عملیات پشتیبانی نمی‌شود.
 - MDRIB_ERR_TIMEDOUT: در حین اجرای عملیات، مهلت پایان یافت^۳.
 - MDRIB_ERR_MAXHANDLE: حداکثر تعداد اتصال‌های مجاز، انجام شده است.
 - MDRIB_WRN_INVALID: یکی از پارامترهای عبور داده شده، معتبر نیست و نادیده گرفته شد.

۱۱-۷ پیشوند برچسب انطباق

پیشوند "C"، در ارتباط با زیربند ۹-۴، باید استفاده شود تا انطباق با این بند را نشان دهد.

۱۲ انقیاد Java API

1 - Pass
2 - I/O error
3 - Time out

API مستقل از انقیاد، مطابق با قواعد زیر، به زبان برنامه نویسی جاوا نگاشت می‌شود. خدمات‌های API روش‌های^۱ کلاس mdrib_handle هستند. انواع داده انقیاد Java API، مبتنی بر زبان برنامه‌نویسی جاوا هستند.

۱-۱۲ نگاشت نوع داده

انواع داده همانند پایین نگاشت می‌شوند:

انقیاد Java API	مستقل از انقیاد
mdrib_handle	mdrib_handle
String	characterstring
reference	pointer
String	str8
String	str16
--	str32
byte, --, short, --, int, --, long, --, --, --	int8, uint8, int16, uint16, int32, uint32, int64, uint64, int128, uint128
time_t	datetime
time_t	duration
String	uri
float, double, --	real32, real64, real80
--	procedure

۲-۱۲ نگاشت پارامتر تابع

پارامترهای API مستقل از انقیاد همانند پارامترهای انقیاد Java API هستند.

۳-۱۲ نگاشت بازگشت تابع

مقادیر بازگشتی همانند پایین نگاشت می‌شوند:

انقیاد Java API	مستقل از انقیاد
int, success=0, failure=-1	state(success,failure)
String	characterstring
String	str8
String	str16
String	str32
byte, --, short, --, int, --, long, --, --, --	int8, uint8, int16, uint16, int32, uint32, int64, uint64, int128, uint128
time_t	datetime
time_t	duration
unsigned char []	uri
float, double, --	real32, real64, real80

۴-۱۲ نگاشت استثنا تابع

استثناهای تابع، همانند پایین نگاشت می‌شوند:

انقیاد Java API	مستقل از انقیاد
exception thrown	bad_conversion

```

/*
 * This class represents a high level connection to
 * a Metadata Registry database. From this connection
 * sessions to the actual data can be retrieved.
 */
package org.iso.mdrib;
import java.util.Properties;
/**
 *
 */
public interface MDRConnection {
/** Opens and creates a session starting
 * at the specified node with the specified options.
 *
 * @returns A valid MDRSession, or null in case of
 * error.
 */
public MDRSession open(String node, Properties options) throws MDRException;
/**
 * Closes the specified session
 */
public void close(MDRSession session) throws MDRException;
/**
 * Authenticates the specified session.
 */
public boolean authenticate(MDRSession session, String authType, Properties options)
throws MDRException;
/**
 * Configures either the current connection, or the session
 */
public void setConfiguration(MDRSession session, String key, String val) throws
MDRException;
/**
 * Returns the configuration of the system as a Properties class
 */
public Properties getConfiguration(MDRSession session) throws MDRException;
}

```

```

public interface MDRConnection {
/** Opens and creates a session starting
 * at the specified node with the specified options.
 *
 * @returns A valid MDRSession, or null in case of

```



```

* error.
*/
public MDRSession open(String node, Properties options) throws MDRException;
/**
* Closes the specified session
*/
public void close(MDRSession session) throws MDRException;
/**
* Authenticates the specified session.
*/
public boolean authenticate(MDRSession session, String authType, Properties options)
throws MDRException;
/**
* Configures either the current connection, or the session
*/
public void setConfiguration(MDRSession session, String key, String val) throws
MDRException;
/**
* Returns the configuration of the system as a Properties class
*/
public Properties getConfiguration(MDRSession session) throws MDRException;
}
* active path (ot absolute path) that match the specified
* mask.
*/
String[] list(String mask) throws MDRException;
/**
* Links an existing object to a new name.
*
*/
void link(String srcIdentifier, String dstIdentifier, String qualifier)
throws MDRException;
/**
* Deletes an object.
*
*/
void delete(String srcIdentifier, String qualifier) throws MDRException;
/**
* Creates an object
*
*/
//void create(String src_identifier) throws MDRException;
/**
* Locks an object.
*
*/
void lock(String srcIdentifier, String qualifier) throws MDRException;
/**
* Unlocks an object.
*

```

```

*/
void unlock(String srcIdentifier, String qualifier) throws MDRException;
/**
 * Copies an object.
 *
 */
void copy(String srcIdentifier, String, dstIdentifier, String qualifier)
throws MDRException;
/**
 * Moves an object.
 *
 */
void move(String srcIdentifier, String, dstIdentifier, String qualifier)
throws MDRException;
/**
 * Retrieves the value of the designated object as a
 * java.math.BigDecimal with full precision.
 */
BigDecimal getBigDecimal(String srcIdentifier) throws MDRException;
/**
 * Retrieves the value of the designated object
 * as a boolean in the Java programming language.
 */
boolean getBoolean(String srcIdentifier) throws MDRException;
/**
 * Retrieves the value of the designated object as a byte
 * in the Java programming language.
 */
byte getByte(String srcIdentifier) throws MDRException;
/**
 * Retrieves the value of the designated object as a byte array
 * in the Java programming language.
 */
byte[] getBytes(String srcIdentifier) throws MDRException;
/**
 * Retrieves the value of the designated object as a byte array
 * in the Java programming language using offsets and maximum lengths.
 *
 * @returns The number of bytes actually transferred,
 */
int getBytes(String srcIdentifier, byte[] dst, int srcOffset, int dstOffset, int count) throws
MDRException;
/**
 * Retrieves the value of the designated object as a java.sql.Date
 * in the Java programming language.
 */
Date getDate(String srcIdentifier) throws MDRException;
/**
 * Retrieves the value of the designated object as a double
 * in the Java programming language.

```

```

*/
double getDouble(String srcIdentifier) throws MDRException;
/**
 * Retrieves the value of the designated object as a float
 * in the Java programming language.
 */
float getFloat(String srcIdentifier) throws MDRException;
/**
 * Retrieves the value of the designated object as an int
 * in the Java programming language.
 */
int getInt(String srcIdentifier) throws MDRException;
/**
 * Retrieves the value of the designated object as a long
 * in the Java programming language.
 */
long getLong(String srcIdentifier) throws MDRException;
/**
 * Retrieves the value of the designated object as a short
 * in the Java programming language.
 */
short getShort(String srcIdentifier) throws MDRException;
/**
 * Retrieves the value of the designated object as a String
 * in the Java programming language.
 */
String getString(String srcIdentifier) throws MDRException;
/**
 * Retrieves the value of the designated object as a java.sql.Time
 * in the Java programming language.
 */
Time getTime(String srcIdentifier) throws MDRException;
/**
 * Retrieves the value of the designated object as a java.sql.Timestamp
 * in the Java programming language.
 */
Timestamp getTimestamp(String srcIdentifier) throws MDRException;
/**
 * Retrieves the value of the designated object as a java.net.URL
 * in the Java programming language.
 */
URL getURL(String srcIdentifier) throws MDRException;
/**
 * Updates the designated object with a java.sql.BigDecimal value.
 */
void updateBigDecimal(String dstIdentifier, BigDecimal x) throws MDRException;
/**
 * Updates the designated object with a boolean value.
 */
void updateBoolean(String dstIdentifier, boolean x) throws MDRException;

```

```

/**
 * Updates the designated object with a byte value.
 */
void updateByte(String dstIdentifier, byte x) throws MDRException;
/**
 * Updates the value of the designated object with the specified byte array
 * in the Java programming language using offsets and maximum lengths.
 *
 * @returns The actual number of bytes transferred.
 */
int updateBytes(byte[] src, String dstIdentifier, int srcOffset, int dstOffset, int count)
throws MDRException;
/**
 * Updates the designated object with a byte array value.
 */
void updateBytes(String dstIdentifier, byte[] x) throws MDRException;
/**
 * Updates the designated object with a java.sql.Date value.
 */
void updateDate(String dstIdentifier, Date x) throws MDRException;
/**
 * Updates the designated object with a double value.
 */
void updateDouble(String dstIdentifier, double x) throws MDRException;
/**
 * Updates the designated object with a float value.
 */
void updateFloat(String dstIdentifier, float x) throws MDRException;
/**
 * Updates the designated object with an int value.
 */
void updateInt(String dstIdentifier, int x) throws MDRException;
/**
 * Updates the designated object with a long value.
 */
void updateLong(String dstIdentifier, long x) throws MDRException;
/**
 * Updates the designated object with a null value. ???
 */
void updateNull(String dstIdentifier) throws MDRException;
/**
 * Updates the designated object with a short value.
 */
void updateShort(String dstIdentifier, short x) throws MDRException;
/**
 * Updates the designated object with a String value.
 */
void updateString(String dstIdentifier, String x) throws MDRException;
/**
 * Updates the designated object with a java.sql.Time value.

```

```

*/
void updateTime(String dstIdentifier, Time x) throws MDRException;
/**
 * Updates the designated object with a java.sql.Timestamp value.
 */
void updateTimestamp(String dstIdentifier, Timestamp x) throws MDRException;
/**
 * Sets the current object node path.
 */
void setCurrentPath(String node) throws MDRException;
/**
 * Retrieves the current object node path.
 */
String getCurrentPath() throws MDRException;
}

```

١٢-٥-٣ كلاس استثناء

خلاصه

```

/*
 * This implements an exception class that can be thrown
 * by any of the interfaces and classes in this package.
 *
 */
package org.iso.mdrib;
/**
 *
 */
public class MDRException extends Exception {
/** Error status: Resource currently not available in specified
mode due to locking mechanism, try again later
*/
public static final short MDRIB_ERR_AGAIN = (short)-1;
// ... Other error codes should be defined here...
/** This contains the error code of the last exception. */
public static short error;
/** Creates a new instance of MDRException */
public MDRException() {
}
/** Creates a new instance of MDRException with the specified status code */
public MDRException(short sw) {
error = sw;
}
/** This is the public method that should be used to throw a MDRException
 * The MDRException shall set its error field indicating the type
 * of error that occurred.
 */
public static void throwIt(short sw) throws MDRException
{
throw new MDRException(sw);
}
}

```

}
}
}

۱۲-۶ خطا/استثنا برگشتی

انواع استثناهای زیر برگردانده می‌شوند:

- MDRIB_ERR_ACCESS: مجوز، عملیات در حال اجرا را نادیده گرفت.
- MDRIB_ERR_AGAIN: در حالت مشخص شده، منابع فعلا موجود نیست به دلیل قفل شدن سازوکار، دوباره امتحان کنید.
- MDRIB_ERR_BADH: مدیریت جلسه یا اتصال، معتبر نیست.
- MDRIB_ERR_INVALID: مقدار یکی از گزینه های خطای نحوی غیرقانونی دارد یا قواعد نحوی پارامتر مقصد معتبر نیست.
- MDRIB_ERR_IO: هنگام اجرای عملیات، خطای ورودی/خروجی^۱ رخ داده است.
- MDRIB_ERR_NOENT: مقصد یا مسیر مشخص شده وجود ندارد.
- MDRIB_ERR_NOMEM: برای بازکردن اتصال، حافظه کافی وجود ندارد.
- MDRIB_ERR_NOOPC: برای ایجاد انباره مشخص شده، فضای کافی وجود ندارد.
- MDRIB_ERR_NOTOBJ: مولفه مسیر به شی معتبر اشاره نمی‌کند.
- MDRIB_ERR_PERM: عملیات به علت نقض دسترسی مجاز نیست.
- MDRIB_ERR_NOSUP: عملیات پشتیبانی نمی‌شود.
- MDRIB_ERR_TIMEDOUT: در حین اجرای عملیات، مهلت پایان یافت.
- MDRIB_ERR_MAXHANDLE: حداکثر تعداد اتصال‌های مجاز، انجام شده است.
- MDRIB_WRN_INVALID: یکی از پارامترهای عبور داده شده، معتبر نیست و نادیده گرفته شد.

۱۲-۷ پیشوند برچسب انطباق

پیشوند "Java"، در ارتباط با زیربند ۹-۴، باید استفاده شود تا انطباق با این بند را نشان دهد.

۱۳ انقیاد ECMAScript API

API مستقل از انقیاد، مطابق با قواعد زیر، به زبان برنامه نویسی ECMAScript نگاشت می‌شود. انواع داده انقیاد ECMAScript API، مبتنی بر زبان برنامه‌نویسی ECMAScript هستند.

۱-۱۳ نگاهت نوع داده

انواع داده همانند پایین نگاهت می‌شوند:

انقیاد ECMAScript API	مستقل از انقیاد
string	mdrib_handle
string	characterstring
reference	pointer
string	str8
string	str16
--	str32
numeric	int8, uint8, int16, uint16, int32, uint32, int64, uint64, int128, uint128
numeric	datetime
numeric	duration
string	uri
numeric	real32, real64, real80
procedure	procedure

۲-۱۳ نگاهت پارامتر تابع

پارامترهای API مستقل از انقیاد همانند پارامترهای انقیاد ECMAScript API هستند.

۳-۱۳ نگاهت بازگشت تابع

مقادیر بازگشتی همانند پایین نگاهت می‌شوند:

انقیاد ECMAScript API	مستقل از انقیاد
numeric, success=0, failure=-1	state(success,failure)
string	characterstring
string	str8
string	str16
--	str32
numeric	int8, uint8, int16, uint16, int32, uint32, int64, uint64, int128, uint128
numeric	datetime
numeric	duration
string	uri
numeric	real32, real64, real80

۴-۱۳ نگاهت استثنا تابع

استثناهای تابع، همانند پایین نگاهت می‌شوند:

انقیاد ECMAScript API	مستقل از انقیاد
mdrib_get_error_number()	bad conversion

۵-۱۳ امضاهای فراخوانی روبه

۱-۵-۱۳ خدمت‌های برقراری جلسه

۱-۱-۵-۱۳ اطلاعات سامانه

خلاصه

mdrib_system_info

```
(  
    $session // session handle  
)
```

۲-۱-۵-۱۳ پیکربندی

خلاصه

mdrib_configure

```
(  
    $session, // session handle  
    $parameter_name, // set parameter "name"  
    $parameter_value // to the value "value"  
)
```

۳-۱-۵-۱۳ اتصال

خلاصه

function mdrib_connect

```
(  
    $target, // repository to connect to  
    $options // connect options  
)
```

۴-۱-۵-۱۳ قطع

خلاصه

function mdrib_disconnect

```
(  
    $session // session handle  
)
```

۵-۱-۵-۱۳ باز

خلاصه

function mdrib_open

```
(  
    $session, // session handle  
    $node, // portion of repository to open  
    $options // connect options  
)
```

۶-۱-۵-۱۳ بسته

خلاصه

function mdrib_close

```
(  
    $session // session handle  
)
```

۲-۵-۱۳ خدمات‌های پارامتر جلسه

۱۳-۵-۲-۱ دریافت مسیر

خلاصه

```
function mdrib_get_path
(
    $session // session handle
)
```

۱۳-۵-۳ خدمات‌های امنیت

۱۳-۵-۳-۱ درخواست اصالت‌سنجی/صدور مجوز

خلاصه

```
function mdrib_request_auth
(
    $session, // session handle
    $auth_type, // auth type
    $auth_options // auth options
)
```

۱۳-۵-۳-۱ پاسخ اصالت‌سنجی/صدور مجوز

خلاصه

```
function mdrib_response_auth
(
    $session, // session handle
    $auth_type, // auth type
    $auth_handler // auth handler function
)
```

۱۳-۵-۴ خدمات‌های انتقال داده

۱۳-۵-۴-۱ دریافت مقدار

خلاصه

```
function mdrib_get_value
(
    $session, // [in] session handle
    $src_identifier, // [in] src object name
    $dst_label_type, // [in] saved label: typeof
    &$dst_label_ptr, // [out] saved label: ptr to
    $dst_type_type, // [in] saved datatype: typeof
    &$dst_type_ptr, // [out] saved datatype: ptr to
    $dst_object_type, // [in] saved value: typeof
    &$dst_object_ptr, // [out] saved value: ptr to
    $dst_proplist_type, // [in] saved proplist: typeof
    &$dst_proplist_ptr, // [out] saved proplist: ptr to
)
```

۱۳-۵-۴-۲ Typed get value

```
function mdrib_get_value_as_str8
(
    $session, // [in] session handle
    $src_identifier // [in] src object name
)
function mdrib_get_value_as_str16
(
    $session, // [in] session handle
    $src_identifier // [in] src object name
)
function mdrib_get_value_as_int8
(
    $session, // [in] session handle
    $src_identifier // [in] src object name
)
function mdrib_get_value_as_uint8
(
    $session, // [in] session handle
    $src_identifier // [in] src object name
)
function mdrib_get_value_as_int16
(
    $session, // [in] session handle
    $src_identifier // [in] src object name
)
function mdrib_get_value_as_uint16
(
    $session, // [in] session handle
    $src_identifier // [in] src object name
)
function mdrib_get_value_as_int32
(
    $session, // [in] session handle
    $src_identifier // [in] src object name
)
function mdrib_get_value_as_uint32
(
    $session, // [in] session handle
    $src_identifier // [in] src object name
)
function mdrib_get_value_as_int64
(
    $session, // [in] session handle
    $src_identifier // [in] src object name
)
function mdrib_get_value_as_uint64
(
    $session, // [in] session handle
    $src_identifier // [in] src object name
```

```

)
function mdrib_get_value_as_datetime
(
    $session, // [in] session handle
    $src_identifier // [in] src object name
)
function mdrib_get_value_as_duration
(
    $session, // [in] session handle
    $src_identifier // [in] src object name
)
function mdrib_get_value_as_uri
(
    $session, // [in] session handle
    $src_identifier // [in] src object name
)
function mdrib_get_value_as_real32
(
    $session, // [in] session handle
    $src_identifier // [in] src object name
)
function mdrib_get_value_as_real64
(
    $session, // [in] session handle
    $src_identifier // [in] src object name
)

```

۳-۴-۵-۱۳ قراردادادن مقدار

خلاصه

```

function mdrib_put_value
(
    $session, // [in] session handle
    $src_identifier, // [in] src object name
    $src_label_type, // [in] assigned label: typeof
    $src_label_ptr, // [in] assigned label: ptr to
    $src_type_type, // [in] assigned datatype: typeof
    $src_type_ptr, // [in] assigned datatype: ptr to
    $src_object_type, // [in] assigned value: typeof
    $src_object_ptr, // [in] assigned value: ptr to
    $src_proplist_type, // [in] assigned proplist: typeof
    $src_proplist_ptr, // [in] assigned proplist: ptr to
)

```

Typed put value ۴-۴-۵-۱۳

خلاصه

```

function mdrib_put_value_as_str8
(
    $session, // [in] session handle
    $dst_identifier, // [in] dst object name

```

```

    $src_value // [in] src value
)
function mdrib_put_value_as_str16
(
    $session, // [in] session handle
    $dst_identifier, // [in] dst object name
    $src_value // [in] src value
)
function mdrib_put_value_as_int8
(
    $session, // [in] session handle
    $dst_identifier, // [in] dst object name
    $src_value // [in] src value
)
function mdrib_put_value_as_uint8
(
    $session, // [in] session handle
    $dst_identifier, // [in] dst object name
    $src_value // [in] src value
)
function mdrib_put_value_as_int16
(
    $session, // [in] session handle
    $dst_identifier, // [in] dst object name
    $src_value // [in] src value
)
function mdrib_put_value_as_uint16
(
    $session, // [in] session handle
    $dst_identifier, // [in] dst object name
    $src_value // [in] src value
)
function mdrib_put_value_as_int32
(
    $session, // [in] session handle
    $dst_identifier, // [in] dst object name
    $src_value // [in] src value
)
function mdrib_put_value_as_uint32
(
    $session, // [in] session handle
    $dst_identifier, // [in] dst object name
    $src_value // [in] src value
)
function mdrib_put_value_as_int64
(
    $session, // [in] session handle
    $dst_identifier, // [in] dst object name
    $src_value // [in] src value
)

```

```

function mdrib_put_value_as_uint64
(
    $session, // [in] session handle
    $dst_identifier // [in] dst object name
    $src_value // [in] src value
)
function mdrib_put_value_as_datetime
(
    $session, // [in] session handle
    $dst_identifier // [in] dst object name
    $src_value // [in] src value
)
function mdrib_put_value_as_duration
(
    $session, // [in] session handle
    $dst_identifier // [in] dst object name
    $src_value // [in] src value
)
function mdrib_put_value_as_uri
(
    $session, // [in] session handle
    $dst_identifier // [in] dst object name
    $src_value // [in] src value
)
mdrib_put_value_as_real32
(
    session, // [in] session handle
    dst_identifier, // [in] dst object name
    src_value // [in] src value
)
mdrib_put_value_as_real64
(
    session, // [in] session handle
    dst_identifier, // [in] dst object name
    src_value // [in] src value
)

```

۵-۵-۱۳ متفرقه

۱-۵-۵-۱۳ خدمت ساختن شی

خلاصه

```

function mdrib_make_object
(
    $session, // [in] session handle
    $src_identifier, // [in] src object name
    $src_label_type, // [in] assigned label: typeof
    $src_label_ptr, // [in] assigned label: ptr to
    $src_type_type, // [in] assigned datatype: typeof
    $src_type_ptr, // [in] assigned datatype: ptr to
    $src_object_type, // [in] assigned value: typeof

```

```
$src_object_ptr, // [in] assigned value: ptr to  
$src_proplist_type, // [in] assigned proplist: typeof  
$src_proplist_ptr, // [in] assigned proplist: ptr to  
)
```

۲-۵-۵-۱۳ خدمت حذف شی

خلاصه

```
function mdrib_remove_object  
(  
    $session, // [in] session handle  
    $src_identifier, // [in] src object name  
)
```

۳-۵-۵-۱۳ خدمت پیوند شی

خلاصه

```
function mdrib_link_object  
(  
    $session, // [in] session handle  
    $src_identifier, // [in] src object name  
    $dst_identifier, // [in] dst object name  
    $link_type // [in] link type: soft, hard  
)
```

۴-۵-۵-۱۳ خدمت یافتن خصوصیت

خلاصه

```
function mdrib_list_object  
(  
    $session, // [in] session handle  
    $src_identifier, // [in] src object name (wildcard)  
    $qualifier // [in] propfind service qualifier  
)
```

۵-۵-۵-۱۳ خدمت قفل کردن شی

خلاصه

```
function mdrib_lock_object  
(  
    $session, // [in] session handle  
    $src_identifier, // [in] src object name  
    $qualifier // [in] lock service qualifier  
)
```

۶-۵-۵-۱۳ خدمت قفل گشودن شی

خلاصه

```
function mdrib_unlock_object  
(  
    $session, // [in] session handle  
    $src_identifier, // [in] src object name  
    $qualifier // [in] unlock service qualifier  
)
```

```
function mdrib_copy_object
(
    $session, // [in] session handle
    $src_identifier, // [in] src object name
    $dst_identifier, // [in] dst object name
    $qualifier // [in] copy service qualifier
)
```

```
function mdrib_move_object
(
    $session, // [in] session handle
    $src_identifier, // [in] src object name
    $dst_identifier, // [in] dst object name
    $qualifier // [in] move service qualifier
)
```

- خطا/استثناهای بازگشتی زیر از طریق متغیر ermo عبور داده می شوند^۱:
- MDRIB_ERR_ACCESS: مجوز، عملیات در حال اجرا را نادیده گرفت.
 - MDRIB_ERR_AGAIN: در حالت مشخص شده، منابع فعلا موجود نیست به دلیل قفل شدن سازوکار، دوباره امتحان کنید.
 - MDRIB_ERR_BADH: مدیریت جلسه یا اتصال، معتبر نیست.
 - MDRIB_ERR_INVALID: مقدار یکی از گزینه های خطای نحوی غیرقانونی دارد یا قواعد نحوی پارامتر مقصد معتبر نیست.
 - MDRIB_ERR_IO: هنگام اجرای عملیات، خطای ورودی/خروجی^۲ رخ داده است.
 - MDRIB_ERR_NOENT: مقصد یا مسیر مشخص شده وجود ندارد.
 - MDRIB_ERR_NOMEM: برای بازکردن اتصال، حافظه کافی وجود ندارد.
 - MDRIB_ERR_NOSPC: برای ایجاد انباره مشخص شده، فضای کافی وجود ندارد.
 - MDRIB_ERR_NOTOBJ: مولفه مسیر به شی معتبر اشاره نمی کند.
 - MDRIB_ERR_PERM: عملیات به علت نقض دسترسی مجاز نیست.
 - MDRIB_ERR_NOSUP: عملیات پشتیبانی نمی شود.
 - MDRIB_ERR_TIMEDOUT: در حین اجرای عملیات، مهلت پایان یافت.

- MDRIB_ERR_MAXHANDLE: حداکثر تعداد اتصال‌های مجاز، انجام شده است.
- MDRIB_WRN_INVALID: یکی از پارامترهای عبور داده شده، معتبر نیست و نادیده گرفته شد.

۷-۱۳ پیشوند برچسب انطباق

پیشوند "ECMAScript"، در ارتباط با زیربند ۹-۴، باید استفاده شود تا انطباق با این بند را نشان دهد.

کتابنامه

- [1] ISO/IEC 8859-1, Information technology — 8-bit single-byte coded graphic character sets — Part 1: Latin alphabet No. 1
- [2] ISO/IEC 10646, Information technology — Universal Coded Character Set (UCS)
- [3] IEC 60559, Binary floating-point arithmetic for microprocessor systems
- [4] IEEE 1596.5, IEEE Standard for Shared-Data Formats Optimized for Scalable Coherent Interface (SCI) Processors
- [5] RFC 4518, Lightweight Directory Access Protocol (LDAP): Internationalized String Preparation, June 2006