GFD-R-P.XXX
DRMAA-WG
drmaa-wg@ogf.org

Daniel Gruber, Univa
September 2014

Distributed Resource Management Application API Version 2 (DRMAA) - Go Language Binding

Status of This Document

OGF Proposed Recommendation (GFD-R-P.XX)

Document Change History

Date	*Notes*
September XXth, 2014	Submission to OGF Editor
October XXth, 2014	Updates from public comment period
November XXth, 2014	Publication as GFD-R-P.XXX

Copyright Notice

Trademark

Abstract

This document describes the Go language binding for the *Distributed Resource Management Application API Version 2 (DRMAA)*. The intended audience for this specification are DRMAA implementors.

[1]Corresponding author

Notational Conventions

In this document, C language elements and definitions are represented in a `fixed-width` font.

The key words “MUST” “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” are to be interpreted as described in RFC 2119 [**?**].

Contents

1 Introduction

The *Distributed Resource Management Application API Version 2 (DRMAA)* specification defines an interface for tightly coupled, but still portable access to the majority of DRM systems. The scope is limited to job submission, job control, reservation management, and retrieval of job and machine monitoring information.

The *DRMAA root specification* [**?**] describes the abstract API concepts and the behavioral rules of a compliant implementation, while this document standardizes the representation of API concepts in the Go programming language.

2 General Design

The mapping of DRMAA IDL constructs to Go follows a set of design principles. Implementation-specific extensions of the DRMAA Go API SHOULD follow these conventions:

- Name spaces of the DRMAA API, as demanded by by the root specification, is realized with the `drmaa2_` Go package name.
- The methods are implemented on structs representing the object name from the IDL specification.
- Return types are references to allocated structs.
- The implementation MAY be based on the C language binding.
- Go comes with its own garbage collector. The implementation SHOULD avoids to call wrappers for the functions defined in the C standard for freeing the data. This usually requires that the methods transforms and copies data available in the C API into its own Go specific counterparts.
- Nevertheless for complete memory management a destructor or an exit function needs to be called.
- Functions are returning additionally an Go type error value. This MAY be casted by the calling function to a drmaa2.Error type which encodes the error reason as a constant, as well as a human readable string representation. The Go implementation MUST provide an return value which fulfills this request (see Error Handling section).
- Go default types are going to be used, i.e. Go maps for dictionaries and slices for lists of return values.
- UNSET values are derived from the underlying C DRMAA values.
- Time values which define a duration are represented with the Go time.Duration type. An UNSET duration is XXX.

2.1 Error Handling

Methods are returning the Go error type. If no error happened then `nil` is returned otherwise the error is set. The error MUST be able to be casted to the `drmaa2.Error` type which is a pointer to a struct consisting of a constant indicating the error reason as well as a string value which describes the error in human readable form. Both values SHOULD provide the same information as the C API calls to `drmaa2_lasterror` and `drmaa2_lasterror_text` (if they are called directly after the function call in the same thread).

2.2 Lists and Dictionaries

Lists are in the Go binding `slices` while Dictionaries are implemented as Go `maps`.

DRMAA interface	Go binding prefix
`DrmaaReflective`	struct methods (Go `embedding`) see below
`SessionManager`	`SessionManager` struct
`JobSession`	`JobSession` struct
`ReservationSession`	`ReservationSession` struct
`MonitoringSession`	`MonitoringSession` struct
`Reservation`	`Reservation` struct
`Job`	`Job` struct
`JobArray`	`ArrayJob` struct
`JobTemplate`	`JobTemplate` struct
`ReservationTemplate`	`ReservationTemplate` struct

Table 1: Mapping of DRMAA interface name to Go structs on which methods are defined

3 Implementation-specific Extensions

The DRMAA root specification allows the product-specific extension of the DRMAA API in a standardized way.

Go method available for an (implementation specific) extensible data structure	Description
`GetExtension(key) (string, error)`	Returns an value of an implementation specific extension.
`SetExtension(key, value string) error`	Sets an extension to a specific value.
`DescribeExtension(string) (string, error)`	Returns a string containing a human readable description of the extension.
`ListExtensions() []string`	Lists all available extension keys for the specific object.

Table 2: Methods available for all extensible DRMAA data structures.

4 Go Public Interface Documentation

The following text shows the complete Go interface description generated by the `go doc`? tool.

DRMAA-compliant Go libraries MUST declare all functions and data structures described here. Implementations MAY add custom parts in adherence to the extensibility principles of this specification and the root specification.

The source file is also available at `http://www.drmaa.org`.

```
PACKAGE

package drmaa2
    import "github.com/dgruber/drmaa2"

CONSTANTS

const (
    JobTemplateType = iota
    JobInfoType
    ReservationTemplateType
    ReservationInfoType
    QueueInfoType
    MachineInfoType
    NotificationType
)
const (
    AdvanceReservation = iota
    ReserveSlots
    Callback
    BulkJobsMaxParallel
    JtEmail
    JtStaging
    JtDeadline
    JtMaxSlots
    JtAccountingId
    RtStartNow
    RtDuration
    RtMachineOS
    RtMachineArch
)
const (
    NewState = iota
    Migrated
    AttributeChange
)
const InfiniteTime = int64(C.DRMAA2_INFINITE_TIME)
    Special timeout value: Wait probably infinitly
const ZeroTime = int64(C.DRMAA2_ZERO_TIME)
    Special timeout value: Don't␣wait


TYPES

type␣ArrayJob␣struct␣{
␣␣␣␣contains␣filtered␣or␣unexported␣fields
}

func␣(aj␣*ArrayJob)␣GetId()␣string
␣␣␣␣Returns␣the␣identifier␣of␣the␣ArrayJob.

func␣(aj␣*ArrayJob)␣GetJobTemplate()␣*JobTemplate
␣␣␣␣Returns␣the␣JobTemplate␣of␣an␣ArrayJob.

func␣(aj␣*ArrayJob)␣GetJobs()␣[]Job
␣␣␣␣Returns␣a␣list␣of␣individual␣jobs␣the␣ArrayJob␣consists␣of.

func␣(aj␣*ArrayJob)␣GetSessionName()␣string
␣␣␣␣Returns␣the␣name␣of␣the␣job␣session␣the␣array␣job␣belongs␣to.

func␣(aj␣*ArrayJob)␣Hold()␣error
␣␣␣␣Sets␣all␣tasks␣of␣an␣ArrayJob␣to␣hold.

func␣(aj␣*ArrayJob)␣Release()␣error
␣␣␣␣Releases␣all␣tasks␣of␣an␣ArrayJob␣from␣hold,␣if␣they␣are␣on␣hold.

func␣(aj␣*ArrayJob)␣Resume()␣error
␣␣␣␣Resumes␣all␣suspended␣tasks␣of␣an␣ArrayJob.

func␣(aj␣*ArrayJob)␣Suspend()␣error
␣␣␣␣Suspends␣all␣running␣tasks␣of␣an␣ArrayJob.

func␣(aj␣*ArrayJob)␣Terminate()␣error
␣␣␣␣Terminates␣(usually␣sends␣a␣SIGKILL)␣all␣tasks␣of␣an␣ArrayJob.

type␣CPU␣int
␣␣␣␣CPU␣architecture␣types

const␣(
␣␣␣␣OtherCPU␣CPU␣=␣iota
␣␣␣␣Alpha
␣␣␣␣ARM
```

```
␣␣␣␣ARM64
␣␣␣␣Cell
␣␣␣␣PA_RISC
␣␣␣␣PA_RISC64
␣␣␣␣IA_64
␣␣␣␣MIPS
␣␣␣␣MIPS64
␣␣␣␣PowerPC
␣␣␣␣PowerPC64
␣␣␣␣SPARC
␣␣␣␣SPARC64
)

func␣(cpu␣CPU)␣String()␣string

type␣CallbackFunction␣func(notification␣Notification)
␣␣␣␣A␣Callback␣is␣a␣function␣which␣works␣on␣the␣notification␣struct.

type␣Capability␣int
␣␣␣␣Capabilities␣are␣optional␣functionalities␣defined␣by␣the␣DRMAA2
␣␣␣␣standard.

type␣Drmaa2Extensible␣interface␣{
␣␣␣␣//␣Lists␣all␣implementation␣specific␣key␣names␣for
␣␣␣␣//␣a␣particular␣DRMAA2␣extensible␣data␣type
␣␣␣␣ListExtensions()␣[]string
␣␣␣␣DescribeExtension(string)␣string
␣␣␣␣SetExtension(string)␣error
␣␣␣␣GetExtension()␣string
}
␣␣␣␣The␣Drmaa2Extensible␣interface␣lists␣all␣functions␣required␣for␣DRMAA2
␣␣␣␣extensible␣data␣structures␣(JobTemplate,␣JobInfo␣etc.).

type␣Error␣struct␣{
␣␣␣␣Message␣string
␣␣␣␣Id␣␣␣␣␣␣ErrorId
}
␣␣␣␣DRMAA2␣error␣(implements␣GO␣Error␣interface).

func␣(ce␣Error)␣Error()␣string
␣␣␣␣The␣DRMAA2␣Error␣implements␣the␣Error␣interface.

func␣(ce␣Error)␣String()␣string
␣␣␣␣Implement␣the␣Stringer␣interface␣for␣an␣drmaa2.Error

type␣ErrorId␣int
␣␣␣␣DRMAA2␣error␣ID

const␣(
␣␣␣␣Success␣ErrorId␣=␣iota
␣␣␣␣DeniedByDrms
␣␣␣␣DrmCommunication
␣␣␣␣TryLater
␣␣␣␣SessionManagement
␣␣␣␣Timeout
␣␣␣␣Internal
␣␣␣␣InvalidArgument
␣␣␣␣InvalidSession
␣␣␣␣InvalidState
␣␣␣␣OutOfResource
␣␣␣␣UnsupportedAttribute
␣␣␣␣UnsupportedOperation
␣␣␣␣ImplementationSpecific
␣␣␣␣LastError
)

type␣Event␣int

type␣Extension␣struct␣{
␣␣␣␣SType␣␣␣␣␣␣␣␣␣␣StructType␣␣␣␣␣␣␣␣//␣Stores␣the␣type␣of␣the␣struct
␣␣␣␣Internal␣␣␣␣␣␣␣unsafe.Pointer␣␣␣␣//␣Enhancmement␣of␣C␣struct
```

```
␣␣␣␣ExtensionList␣map[string]string␣//␣stores␣the␣extension␣requests␣as␣string
}
␣␣␣␣Extension␣struct␣which␣is␣embedded␣in␣DRMAA2␣objects␣which␣are
␣␣␣␣extensible.

func␣(e␣*Extension)␣GetExtension(extension␣string)␣(string,␣error)
␣␣␣␣For␣all␣types␣which␣embedds␣the␣Extension␣struct␣(JobTemplate␣etc.)

type␣Job␣struct␣{
␣␣␣␣//␣contains␣filtered␣or␣unexported␣fields
}

func␣(job␣*Job)␣GetId()␣string

func␣(job␣*Job)␣GetJobInfo()␣(*JobInfo,␣error)
␣␣␣␣Creates␣a␣JobInfo␣object␣from␣the␣job␣containing␣more␣detailed
␣␣␣␣information␣about␣the␣job.

func␣(job␣*Job)␣GetJobTemplate()␣(*JobTemplate,␣error)
␣␣␣␣Returns␣the␣JobTemplate␣used␣to␣submit␣the␣job.

func␣(job␣*Job)␣GetSessionName()␣string

func␣(job␣*Job)␣GetState()␣JobState
␣␣␣␣Returns␣the␣current␣JobState␣of␣the␣job.

func␣(job␣*Job)␣Hold()␣error
␣␣␣␣Puts␣the␣job␣into␣an␣hold␣state␣so␣that␣it␣is␣not␣scheduled.␣If␣the␣job
␣␣␣␣is␣already␣running␣it␣continues␣to␣run␣and␣the␣hold␣state␣becomes␣only
␣␣␣␣effectice␣when␣the␣job␣is␣rescheduled.

func␣(job␣*Job)␣Release()␣error
␣␣␣␣Releases␣the␣job␣from␣hold␣state␣so␣that␣it␣will␣be␣schedulable.

func␣(job␣*Job)␣Resume()␣error
␣␣␣␣Resumes␣a␣job␣/␣process␣to␣be␣runnable␣again.

func␣(job␣*Job)␣Suspend()␣error
␣␣␣␣Stops␣a␣job␣/␣process␣from␣beeing␣executed.

func␣(job␣*Job)␣WaitStarted(timeout␣int64)␣error
␣␣␣␣Blocking␣wait␣until␣the␣job␣is␣started.␣The␣timeout␣prefents␣that␣the
␣␣␣␣call␣is␣blocking␣endlessly.␣Special␣timeouts␣are␣available␣by␣the
␣␣␣␣constants␣InfiniteTime␣and␣ZeroTime.

func␣(job␣*Job)␣WaitTerminated(timeout␣int64)␣error
␣␣␣␣Waits␣until␣the␣job␣goes␣into␣one␣of␣the␣finished␣states.␣The␣timeout
␣␣␣␣specifies␣the␣maximum␣time␣to␣wait.␣If␣no␣timeout␣is␣required␣use␣the
␣␣␣␣constant␣InfiniteTime.

type␣JobInfo␣struct␣{
␣␣␣␣//␣reference␣to␣the␣void*␣pointer␣which
␣␣␣␣//␣is␣used␣for␣extensions
␣␣␣␣Extension
␣␣␣␣Id␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣string
␣␣␣␣ExitStatus␣␣␣␣␣␣␣␣int
␣␣␣␣TerminatingSignal␣string
␣␣␣␣Annotation␣␣␣␣␣␣␣␣string
␣␣␣␣State␣␣␣␣␣␣␣␣␣␣␣␣␣JobState
␣␣␣␣SubState␣␣␣␣␣␣␣␣␣␣string
␣␣␣␣AllocatedMachines␣[]string
␣␣␣␣SubmissionMachine␣string
␣␣␣␣JobOwner␣␣␣␣␣␣␣␣␣␣string
␣␣␣␣Slots␣␣␣␣␣␣␣␣␣␣␣␣␣int64
␣␣␣␣QueueName␣␣␣␣␣␣␣␣␣string
␣␣␣␣WallclockTime␣␣␣␣␣time.Duration
␣␣␣␣CPUTime␣␣␣␣␣␣␣␣␣␣␣int64
␣␣␣␣SubmissionTime␣␣␣␣time.Time
␣␣␣␣DispatchTime␣␣␣␣␣␣time.Time
␣␣␣␣FinishTime␣␣␣␣␣␣␣␣time.Time
}
```

```
func␣(structType␣*JobInfo)␣ListExtensions()␣[]string
␣␣␣␣Returns␣a␣string␣list␣containing␣all␣implementation␣specific␣extensions
␣␣␣␣of␣the␣JobInfo␣object.

func␣(ji␣*JobInfo)␣SetExtension(extension,␣value␣string)␣error

type␣JobSession␣struct␣{
␣␣␣␣Name␣string␣//␣public␣name␣of␣job␣session
␣␣␣␣//␣contains␣filtered␣or␣unexported␣fields
}

func␣(js␣*JobSession)␣Close()␣error
␣␣␣␣Closes␣an␣open␣JobSession.

func␣(js␣*JobSession)␣GetContact()␣(string,␣error)
␣␣␣␣Returns␣the␣contact␣string␣of␣the␣DRM␣session.

func␣(js␣*JobSession)␣GetJobArray(id␣string)␣(*ArrayJob,␣error)
␣␣␣␣Returns␣a␣reference␣to␣an␣existing␣ArrayJob␣based␣on␣the␣given␣job␣id.
␣␣␣␣In␣case␣of␣an␣error␣the␣error␣return␣value␣is␣set␣to␣!=␣nil.

func␣(js␣*JobSession)␣GetJobCategories()␣([]string,␣error)
␣␣␣␣Returns␣all␣job␣categories␣specified␣for␣the␣job␣session.

func␣(js␣*JobSession)␣GetJobs()␣([]Job,␣error)
␣␣␣␣Returns␣a␣list␣of␣all␣jobs␣currently␣running␣in␣the␣given␣JobSession.

func␣(js␣*JobSession)␣GetSessionName()␣(string,␣error)

func␣(js␣*JobSession)␣RunBulkJobs(jt␣JobTemplate,␣begin␣int,␣end␣int,␣step␣int,␣maxParallel␣int)␣(*ArrayJob,␣error)
␣␣␣␣Submits␣a␣JobTemplate␣to␣the␣cluster␣as␣an␣array␣job␣(multiple␣instances
␣␣␣␣of␣the␣same␣job,␣not␣neccessarly␣running␣a␣the␣same␣point␣in␣time).␣It
␣␣␣␣requires␣a␣JobTemplate␣filled␣out␣at␣least␣with␣a␣RemoteCommand.␣The
␣␣␣␣begin,␣end␣and␣step␣parameters␣specifying␣how␣many␣array␣job␣instances
␣␣␣␣are␣submitted␣and␣how␣the␣instances␣are␣numbered␣(1,10,1␣denotes␣10
␣␣␣␣array␣job␣instances␣numbered␣from␣1␣to␣10).␣The␣maxParallel␣parameter
␣␣␣␣specifies␣how␣many␣of␣the␣array␣job␣instances␣should␣run␣at␣parallel␣as
␣␣␣␣maximum␣(when␣resources␣are␣contrainted␣then␣less␣instances␣could␣run).

func␣(js␣*JobSession)␣RunJob(jt␣JobTemplate)␣(*Job,␣error)
␣␣␣␣Submits␣a␣job␣based␣on␣the␣parameters␣specified␣in␣the␣JobTemplate␣in
␣␣␣␣the␣cluster.␣In␣case␣of␣success␣it␣returns␣a␣pointer␣to␣a␣Job␣element,
␣␣␣␣which␣can␣be␣used␣for␣further␣processing.␣In␣case␣of␣an␣error␣the␣error
␣␣␣␣return␣value␣is␣set.

func␣(js␣*JobSession)␣WaitAnyStarted(jobs␣[]Job,␣timeout␣int64)␣(*Job,␣error)
␣␣␣␣Waits␣until␣any␣of␣the␣given␣jobs␣is␣started␣(usually␣in␣running␣state).
␣␣␣␣The␣timeout␣determines␣after␣how␣many␣seconds␣the␣method␣should␣abort,
␣␣␣␣even␣when␣none␣of␣the␣given␣jobs␣was␣started.␣Special␣timeout␣values␣are
␣␣␣␣InfiniteTime␣and␣ZeroTime.

func␣(js␣*JobSession)␣WaitAnyTerminated(jobs␣[]Job,␣timeout␣int64)␣(*Job,␣error)
␣␣␣␣Waits␣until␣any␣of␣the␣given␣jobs␣is␣finished.␣The␣timeout␣determines
␣␣␣␣after␣how␣many␣seconds␣the␣method␣should␣abort,␣even␣when␣none␣of␣the
␣␣␣␣given␣jobs␣is␣finished.␣Sepecial␣timeout␣values␣are␣InfiniteTime␣and
␣␣␣␣ZeroTime.

type␣JobState␣int
␣␣␣␣Job␣States

const␣(
␣␣␣␣Undetermined␣JobState␣=␣iota
␣␣␣␣Queued
␣␣␣␣QueuedHeld
␣␣␣␣Running
␣␣␣␣Suspended
␣␣␣␣Requeued
␣␣␣␣RequeuedHeld
␣␣␣␣Done
␣␣␣␣Failed
```

```
)

func␣(js␣JobState)␣String()␣string
␣␣␣␣Implements␣the␣Stringer␣interface

type␣JobTemplate␣struct␣{
␣␣␣␣Extension
␣␣␣␣RemoteCommand␣␣␣␣␣string
␣␣␣␣Args␣␣␣␣␣␣␣␣␣␣␣␣␣␣[]string
␣␣␣␣SubmitAsHold␣␣␣␣␣␣bool
␣␣␣␣ReRunnable␣␣␣␣␣␣␣␣bool
␣␣␣␣JobEnvironment␣␣␣␣map[string]string
␣␣␣␣WorkingDirectory␣␣string
␣␣␣␣JobCategory␣␣␣␣␣␣␣string
␣␣␣␣Email␣␣␣␣␣␣␣␣␣␣␣␣␣[]string
␣␣␣␣EmailOnStarted␣␣␣␣bool
␣␣␣␣EmailOnTerminated␣bool
␣␣␣␣JobName␣␣␣␣␣␣␣␣␣␣␣string
␣␣␣␣InputPath␣␣␣␣␣␣␣␣␣string
␣␣␣␣OutputPath␣␣␣␣␣␣␣␣string
␣␣␣␣ErrorPath␣␣␣␣␣␣␣␣␣string
␣␣␣␣JoinFiles␣␣␣␣␣␣␣␣␣bool
␣␣␣␣ReservationId␣␣␣␣␣string
␣␣␣␣QueueName␣␣␣␣␣␣␣␣␣string
␣␣␣␣MinSlots␣␣␣␣␣␣␣␣␣␣int64
␣␣␣␣MaxSlots␣␣␣␣␣␣␣␣␣␣int64
␣␣␣␣Priority␣␣␣␣␣␣␣␣␣␣int64
␣␣␣␣CandidateMachines␣[]string
␣␣␣␣MinPhysMemory␣␣␣␣␣int64
␣␣␣␣MachineOs␣␣␣␣␣␣␣␣␣string
␣␣␣␣MachineArch␣␣␣␣␣␣␣string
␣␣␣␣StartTime␣␣␣␣␣␣␣␣␣time.Time
␣␣␣␣DeadlineTime␣␣␣␣␣␣time.Time
␣␣␣␣StageInFiles␣␣␣␣␣␣map[string]string
␣␣␣␣StageOutFiles␣␣␣␣␣map[string]string
␣␣␣␣ResourceLimits␣␣␣␣map[string]string
␣␣␣␣AccountingId␣␣␣␣␣␣string
}

func␣(jt␣*JobTemplate)␣DescribeExtension(extensionName␣string)␣(string,␣error)
␣␣␣␣Returns␣the␣description␣of␣an␣implementation␣specific␣JobTemplate
␣␣␣␣extension␣as␣a␣string.

func␣(structType␣*JobTemplate)␣ListExtensions()␣[]string
␣␣␣␣Returns␣a␣string␣list␣containing␣all␣implementation␣specific␣extensions
␣␣␣␣of␣the␣JobTemplate␣object.

func␣(jt␣*JobTemplate)␣SetExtension(extension,␣value␣string)␣error

type␣Machine␣struct␣{
␣␣␣␣Extension
␣␣␣␣Name␣␣␣␣␣␣␣␣␣␣␣string
␣␣␣␣Available␣␣␣␣␣␣bool
␣␣␣␣Sockets␣␣␣␣␣␣␣␣int64
␣␣␣␣CoresPerSocket␣int64
␣␣␣␣ThreadsPerCore␣int64
␣␣␣␣Load␣␣␣␣␣␣␣␣␣␣␣float64
␣␣␣␣PhysicalMemory␣int64
␣␣␣␣VirtualMemory␣␣int64
␣␣␣␣Architecture␣␣␣CPU
␣␣␣␣OSVersion␣␣␣␣␣␣Version
␣␣␣␣OS␣␣␣␣␣␣␣␣␣␣␣␣␣OS
}

func␣(structType␣*Machine)␣ListExtensions()␣[]string
␣␣␣␣Returns␣a␣string␣list␣containing␣all␣implementation␣specific␣extensions
␣␣␣␣of␣the␣Machine␣object.␣/

func␣(m␣*Machine)␣SetExtension(extension,␣value␣string)␣error

type␣MonitoringSession␣struct␣{
```

```
␣␣␣␣//␣contains␣filtered␣or␣unexported␣fields
}

func␣(ms␣*MonitoringSession)␣CloseMonitoringSession()␣error
␣␣␣␣Closes␣the␣MonitoringSession.

func␣(ms␣*MonitoringSession)␣GetAllJobs()␣(jobs␣[]Job,␣err␣error)
␣␣␣␣Returns␣a␣slice␣of␣jobs␣currently␣visible␣in␣the␣monitoring␣session.

func␣(ms␣*MonitoringSession)␣GetAllMachines()␣(machines␣[]Machine,␣err␣error)
␣␣␣␣Returns␣a␣list␣of␣all␣machines␣configured␣in␣cluster.

func␣(ms␣*MonitoringSession)␣GetAllQueues()␣(queues␣[]Queue,␣err␣error)
␣␣␣␣Returns␣all␣queues␣configured␣in␣the␣cluster.

func␣(ms␣*MonitoringSession)␣GetAllReservations()␣(reservations␣[]Reservation,␣err␣error)

type␣Notification␣struct␣{
␣␣␣␣Evt␣␣␣␣␣␣␣␣␣Event
␣␣␣␣JobId␣␣␣␣␣␣␣string
␣␣␣␣SessionName␣string
␣␣␣␣State␣␣␣␣␣␣␣JobState
}

type␣OS␣int
␣␣␣␣Operating␣System␣type

const␣(
␣␣␣␣OtherOS␣OS␣=␣iota
␣␣␣␣AIX
␣␣␣␣BSD
␣␣␣␣Linux
␣␣␣␣HPUX
␣␣␣␣IRIX
␣␣␣␣MacOS
␣␣␣␣SunOS
␣␣␣␣TRU64
␣␣␣␣UnixWare
␣␣␣␣Win
␣␣␣␣WinNT
)

func␣(os␣OS)␣String()␣string
␣␣␣␣An␣OS␣struct␣needs␣to␣be␣printable.

type␣Queue␣struct␣{
␣␣␣␣Extension
␣␣␣␣Name␣string
}

func␣(structType␣*Queue)␣ListExtensions()␣[]string
␣␣␣␣Returns␣a␣string␣list␣containing␣all␣implementation␣specific␣extensions
␣␣␣␣of␣the␣Queue␣object.

func␣(q␣*Queue)␣SetExtension(extension,␣value␣string)␣error

type␣Reservation␣struct␣{
␣␣␣␣SessionName␣␣␣string
␣␣␣␣Contact␣␣␣␣␣␣␣string
␣␣␣␣Template␣␣␣␣␣␣ReservationTemplate
␣␣␣␣ReservationId␣string
}

func␣(r␣*Reservation)␣GetId()␣(string,␣error)
␣␣␣␣Returns␣the␣advance␣reservation␣id

func␣(r␣*Reservation)␣GetInfo()␣(*ReservationInfo,␣error)
␣␣␣␣␣Returns␣the␣reservation␣info␣object␣of␣the␣reservation

func␣(r␣*Reservation)␣GetSessionName()␣(string,␣error)
␣␣␣␣␣Returns␣the␣name␣of␣the␣reservation
```

```
func␣(r␣*Reservation)␣GetTemplate()␣(*ReservationTemplate,␣error)
␣␣␣␣␣Returns␣the␣reservation␣template␣of␣the␣reservation

func␣(r␣*Reservation)␣Terminate()␣error
␣␣␣␣␣Cancels␣an␣advance␣reservation

type␣ReservationInfo␣struct␣{
␣␣␣␣ReservationId␣␣␣␣␣␣␣␣string
␣␣␣␣ReservationName␣␣␣␣␣␣string
␣␣␣␣ReservationStartTime␣time.Time
␣␣␣␣ReservationEndTime␣␣␣time.Time
␣␣␣␣ACL␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣[]string
␣␣␣␣ReservedSlots␣␣␣␣␣␣␣␣int64
␣␣␣␣ReservedMachines␣␣␣␣␣[]string
}

type␣ReservationSession␣struct␣{
␣␣␣␣//␣contains␣filtered␣or␣unexported␣fields
}

func␣(rs␣*ReservationSession)␣Close()␣error
␣␣␣␣Closes␣an␣open␣ReservationSession.

func␣(rs␣*ReservationSession)␣GetContact()␣(string,␣error)
␣␣␣␣Returns␣the␣contact␣string␣of␣the␣reservation␣session.

func␣(rs␣*ReservationSession)␣GetReservation(rid␣string)␣(*Reservation,␣error)
␣␣␣␣␣Returns␣a␣reservation␣object␣based␣on␣the␣AR␣id

func␣(rs␣*ReservationSession)␣GetReservations()␣([]Reservation,␣error)
␣␣␣␣␣Returns␣a␣list␣of␣available␣advance␣reservations

func␣(rs␣*ReservationSession)␣GetSessionName()␣(string,␣error)
␣␣␣␣␣Returns␣the␣name␣of␣the␣reservation␣session

func␣(rs␣*ReservationSession)␣RequestReservation(rtemplate␣ReservationTemplate)␣(*Reservation,␣error)
␣␣␣␣␣Allocates␣an␣advance␣reservation␣based␣on␣the␣reservation
␣␣␣␣template

type␣ReservationTemplate␣struct␣{
␣␣␣␣Extension
␣␣␣␣Name␣␣␣␣␣␣␣␣␣␣␣␣␣␣string
␣␣␣␣StartTime␣␣␣␣␣␣␣␣␣time.Time
␣␣␣␣EndTime␣␣␣␣␣␣␣␣␣␣␣time.Time
␣␣␣␣Duration␣␣␣␣␣␣␣␣␣␣time.Duration
␣␣␣␣MinSlots␣␣␣␣␣␣␣␣␣␣int64
␣␣␣␣MaxSlots␣␣␣␣␣␣␣␣␣␣int64
␣␣␣␣JobCategory␣␣␣␣␣␣␣string
␣␣␣␣UsersACL␣␣␣␣␣␣␣␣␣␣[]string
␣␣␣␣CandidateMachines␣[]string
␣␣␣␣MinPhysMemory␣␣␣␣␣int64
␣␣␣␣MachineOs␣␣␣␣␣␣␣␣␣string
␣␣␣␣MachineArch␣␣␣␣␣␣␣string
}

type␣SessionManager␣struct␣{
}
␣␣␣␣␣A␣Create␣Method␣which␣initializes␣the␣values␣and␣also␣does
␣␣␣␣initialization␣about␣capabilities,␣versions␣etc.␣?!?

func␣(sm␣*SessionManager)␣CreateJobSession(sessionName,␣contact␣string)␣(*JobSession,␣error)
␣␣␣␣Creates␣a␣new␣persistent␣job␣session␣and␣opens␣it.

func␣(sm␣*SessionManager)␣CreateReservationSession(sessionName,␣contact␣string)␣(rs␣*ReservationSession,␣err␣error)
␣␣␣␣Creates␣a␣ReservationSession␣by␣name␣and␣contact␣string.

func␣(sm␣*SessionManager)␣DestroyJobSession(sessionName␣string)␣error
␣␣␣␣Destroys␣a␣job␣session␣by␣name.

func␣(sm␣*SessionManager)␣DestroyReservationSession(sessionName␣string)␣error
```

```
␣␣␣␣Destroys␣a␣reservation␣by␣name.

func␣(sm␣*SessionManager)␣GetDrmsName()␣(string,␣error)
␣␣␣␣Returns␣the␣name␣of␣the␣Distributed␣Resource␣Management␣System.

func␣(sm␣*SessionManager)␣GetDrmsVersion()␣(*Version,␣error)
␣␣␣␣Returns␣the␣version␣of␣the␣Distributed␣Resource␣Management␣System.

func␣(sm␣*SessionManager)␣GetJobSessionNames()␣([]string,␣error)
␣␣␣␣Returns␣all␣job␣sessions␣accessable␣to␣the␣user.

func␣(sm␣*SessionManager)␣GetReservationSessionNames()␣([]string,␣error)
␣␣␣␣Returns␣all␣reservation␣sessions␣accessable␣to␣the␣user.

func␣(sm␣*SessionManager)␣OpenJobSession(sessionName␣string)␣(js␣*JobSession,␣err␣error)

func␣(sm␣*SessionManager)␣OpenMonitoringSession(sessionName␣string)␣(*MonitoringSession,␣error)
␣␣␣␣Opens␣a␣MonitoringSession␣by␣name.␣Usually␣the␣name␣is␣ignored.

func␣(sm␣*SessionManager)␣OpenReservationSession(name␣string)␣(rs␣ReservationSession,␣err␣error)
␣␣␣␣Opens␣an␣existing␣ReservationSession␣by␣name.

func␣(sm␣*SessionManager)␣RegisterEventNotification(callback␣*CallbackFunction)␣error
␣␣␣␣␣This␣function␣needs␣to␣store␣a␣CallbackFunction␣and␣calls␣it
␣␣␣␣whenever␣a␣event␣occured.

func␣(sm␣*SessionManager)␣Supports(c␣Capability)␣bool
␣␣␣␣Checks␣whether␣the␣DRMAA2␣implementation␣supports␣an␣optional
␣␣␣␣functionality␣or␣not.

type␣StructType␣int
␣␣␣␣In␣order␣to␣make␣extension␣functions␣dependend␣from␣the␣type␣of␣the
␣␣␣␣struct␣we␣need␣to␣store␣the␣type␣somewhere.

type␣Version␣struct␣{
␣␣␣␣Major␣string
␣␣␣␣Minor␣string
}

func␣(v␣*Version)␣String()␣string
```

5 Contributors

Daniel Gruber
Univa GmbH
c/o Rüter und Partner
Prielmayerstr. 3
80335 München, Germany
Email: dgruber@univa.com

6 Intellectual Property Statement

7 Disclaimer

8 Full Copyright Notice

9 References