



Oscar Bundle Repository

[Where is Oscar?](#) | [What is OBR?](#) | [Bundles](#) | [Submitting Bundles](#) | [Contact](#)

News

- Released updates to the Bundle Repository and Shell Plugin bundles; refer to [Bundle Repository change log](#) and the [Shell Plugin change log](#) for more details. These updates are accessible using the `obr update` command. (July 1st, 2004)
- Released two updates for Bundle Repository and Shell Plugin to fix some minor issues; these updates are accessible using the `obr update` command. (June 20th, 2004)
- New OBR web site introduced. (June 13th, 2004)
- Oscar has moved to a new web host at <http://oscar.objectweb.org>. (June 13th, 2004)

[\(News archive\)](#)

What is OBR?

OBR is an incubator and repository for OSGi bundles. For those who are unaware, the [OSGi Alliance](#) is a consortium working on defining standards for delivering and managing dynamically downloadable services into networked environments. The OSGi Alliance has [defined](#) a dynamically extensible framework for this purpose, which supports the dynamic deployment and execution of components, called bundles. The OSGi framework provides an excellent platform for building dynamically extensible applications.

OBR has two two main goals:

- Provide a repository of useful and/or didactic bundles that can be easily deployed into existing OSGi frameworks.
- Promote a community effort around bundle creation by increasing the visibility of individual bundles.

The main approach that OBR uses to achieve these goals is to provide simple access mechanisms for the bundles in the repository. Consequently, there are multiple ways to access the repository bundles:

- **Web access** - all bundles in the OBR repository are accessible via this web page.
- **Programmatic access** - the [Bundle Repository](#) bundle provides an OSGi service for dynamically deploying repository bundles and the transitive closure of their deployment dependencies into an executing OSGi framework.
- **Interactive access** - the [Bundle Repository](#) bundle also provides a shell command for Oscar's [Shell Service](#) bundle, so if you are using Oscar you can interactively type OBR commands at Oscar's shell prompt. Additionally, the [Shell Plugin](#) bundle provides a GUI interface for the [Shell GUI](#) bundle.

While many of the bundles were developed for the [Oscar](#) OSGi framework implementation, they should work with other implementations, such as [Knopflerfish](#).

Bundles

The following is a complete list of all bundles in the OBR repository. Not all bundles are hosted on this site and individual bundles have varying licenses; please consult the individual bundle documentation for details. The source for these bundles is an excellent way to learn how to implement bundles, but you may also want to review the [bundle tutorial](#).

- **Bundle Repository** ([docs](#) | [bundle](#) | [source](#)) - A bundle repository service for Oscar.
- **Content Handler** ([docs](#) | [bundle](#) | [source](#)) - An example content handler for URL Handlers service.
- **Data Stream Handler** ([docs](#) | [bundle](#) | [source](#)) - An example stream handler for URL Handlers service.
- **Device Manager** ([docs](#) | [bundle](#) | [source](#)) - Manages the core service for the device access specification
- **Handler Test** ([docs](#) | [bundle](#) | [source](#)) - Tests the OSGi URL Handlers service.
- **HTTP Admin** ([docs](#) | [bundle](#) | [source](#)) - A simple HTTP-based bundle administrator.
- **HTTP Driver Locator** ([docs](#) | [bundle](#) | [source](#)) - Implements the Driver locator service through a simple http access
- **HTTP Driver Resource** ([docs](#) | [bundle](#) | [source](#)) - Registers a specific context for drivers downloading through an httpserver
- **HTTP Service** ([docs](#) | [bundle](#) | [source](#)) - An implementation of the OSGi HTTP Service using Jetty.
- **HTTP Test** ([docs](#) | [bundle](#) | [source](#)) - Simple test bundle for the HTTP Service.
- **JMX Bundle** ([docs](#) | [bundle](#) | [source](#)) - A bundle that starts a JMX agent (mx4j).
- **Log Reader** ([docs](#) | [bundle](#) | [source](#)) - Reads log events from log service
- **Log Service** ([docs](#) | [bundle](#) | [source](#)) - OSGi compliant log service.
- **MBean Factory** ([docs](#) | [bundle](#) | [source](#)) - A service that manages MBeans representing services.
- **MBeanPsCommand** ([docs](#) | [bundle](#) | [source](#)) - A shell command service to lists services.
- **Messenger** ([docs](#) | [bundle](#) | [source](#)) - A simple instant messenger.
- **Module Manager** ([docs](#) | [bundle](#) | [source](#)) - Manages linux modules through insmod/rmmod calls
- **Notification Service** ([docs](#) | [bundle](#) | [source](#)) - Notification service based on the Siena event system.
- **OSGi Service** ([docs](#) | [bundle](#) | [source](#)) - Library for all org.osgi.service packages.
- **OSGi Util** ([docs](#) | [bundle](#) | [source](#)) - Library for all org.osgi.util packages.
- **PCMCIA Device service** ([docs](#) | [bundle](#) | [source](#)) - Defines a simple pcmcia device bridge (a base driver)
- **Permission Manager** ([docs](#) | [bundle](#) | [source](#)) - A simple bundle to modify permissions.
- **Service Binder** ([docs](#) | [bundle](#) | [source](#)) - Mechanism that automates service dependency management.
- **Service Introspector** ([docs](#) | [bundle](#) | [source](#)) - A service for introspecting services.
- **Service Lookup** ([docs](#) | [bundle](#) | [source](#)) - A simple bundle to test service lookup permissions.
- **Service Notifier** ([docs](#) | [bundle](#) | [source](#)) - A service that observes service life cycles.
- **Service Register** ([docs](#) | [bundle](#) | [source](#)) - A simple bundle to test service registration permissions.
- **Servlet** ([docs](#) | [bundle](#) | [source](#)) - A library bundle javax.servlet packages.
- **Shell Extra** ([docs](#) | [bundle](#) | [source](#)) - Extra shell commands for Oscar's shell.
- **Shell GUI** ([docs](#) | [bundle](#) | [source](#)) - Simple GUI shell for Oscar (use with Shell Plugin).
- **Shell MBean** ([docs](#) | [bundle](#) | [source](#)) - An MBean to access the shell service.

- **Shell Plugin** ([docs](#) | [bundle](#) | [source](#)) - Bundle list, console, and OBR plugins for Shell GUI.
- **Shell Service** ([docs](#) | [bundle](#) | [source](#)) - Simple shell service for interacting with Oscar.
- **Shell TUI** ([docs](#) | [bundle](#) | [source](#)) - Simple text-based shell for Oscar.
- **Simple** ([docs](#) | [bundle](#) | [source](#)) - Example bundle that demonstrates OSGi features.
- **Table Layout** ([docs](#) | [bundle](#) | [source](#)) - Table layout library bundle.
- **Telnet Service** ([docs](#) | [bundle](#) | [source](#)) - General telnet daemon service bundle for OSGi.
- **Tiny Shell Component** ([docs](#) | [bundle](#) | [source](#)) - Tiny components that implement a GUI interface for Oscar
- **Tiny Shell GUI** ([docs](#) | [bundle](#) | [source](#)) - A Simple Tiny GUI for Oscar
- **URL Handlers** ([docs](#) | [bundle](#) | [source](#)) - Implementation of the OSGi Handlers service.
- **Wire Admin** ([docs](#) | [bundle](#) | [source](#)) - Wire Admin service implementation.
- **Wire Admin Command** ([docs](#) | [bundle](#) | [source](#)) - Wire Admin Service commands for Oscar shell.
- **XML RPC** ([docs](#) | [bundle](#) | [source](#)) - General XML/RPC server bundle.

Submitting Bundles

Currently, there is no automated way to submit a bundle to OBR. To submit a bundle to OBR, send its meta-data, encoded in the XML format used by OBR, to [Richard S. Hall](#); see the [repository.xml](#) for examples. Refer to the [bundle source packaging](#) section of the Bundle Repository bundle documentation. In the future, tools could be developed to simplify this process and make it possible to generate bundle manifest and OBR meta-data formats.

Note: It is not necessary to physically host your bundle on this site. It is possible to just add your bundle's meta-data to OBR and have the meta-data URLs point to your own host.

Contact

Any questions? Want to include your bundle? Want to contribute? Contact [Richard S. Hall](#).

Contributors

[Rob Walker](#) from [Ascert](#).

[Stephane Frenot](#) from [l'INSA de Lyon](#).

[Humberto Cervantes](#) from [LSR-IMAG](#).

[Didier Donsez](#) from [LSR-IMAG](#).

[Stephane Chomat](#) from [LSR-IMAG](#).

