THE ACE ORB | AN OCI REDISTRIBUTION

# Getting Started Guide:
# ACE, TAO, and OpenDDS
# VxWorks 7 Marketplace Packages

OCI | WE ARE SOFTWARE ENGINEERS.

**OCI ACE+TAO version 2.2a_p12 [http://theaceorb.com](http://theaceorb.com)**
**OpenDDS version 3.12 [http://opendds.org](http://opendds.org)**
**Tested with Wind River VxWorks 7 version August 2017 (SR0510)**
**By Object Computing, Inc. [http://objectcomputing.com](http://objectcomputing.com)**
**Document Revision 11/7/2017**

## Selecting and Installing a Marketplace Package

The three products ACE, TAO, and OpenDDS each have their own VxWorks 7 Marketplace Packages (https://marketplace.windriver.com). Since TAO depends on ACE, the TAO Marketplace Package includes ACE—you do not have to install the ACE Marketplace Package if you're installing the TAO Marketplace Package. Similarly, OpenDDS depends on both ACE and TAO. If you're going to use OpenDDS, just install the OpenDDS Marketplace Package and all three products will be available in your VxWorks 7 installation. Therefore it is not necessary to have more than one of the three Marketplace Packages installed, though it will not cause not an error to do so (for example, installing TAO and then adding OpenDDS later). On Windows host systems, also install ActiveState Perl since the VSB build process for these layers runs Perl scripts.

## RPM Packages and Their Contents

The various Marketplace Packages ("M.P." in the table below) each provide a number of RPM packages for installation. See the VxWorks 7 document "Layers and Package Management" for more details on how VxWorks makes use of RPMs. We have created the following RPMs:

| RPM name | Version | In ACE M.P.? | In TAO M.P.? | In OpenDDS M.P.? |
|---|---|---|---|---|
| oci_ace | 6.2.12.0 | ✓ | ✓ | ✓ |
| oci_tao_host_linux | 2.2.12.0 | | ✓ | ✓ |
| oci_tao_host_windows | 2.2.12.0 | | ✓ | ✓ |
| oci_tao | 2.2.12.0 | | ✓ | ✓ |
| oci_opendds_host_linux | 3.12.0.0 | | | ✓ |
| oci_opendds_host_windows | 3.12.0.0 | | | ✓ |
| oci_opendds | 3.12.0.0 | | | ✓ |

The RPMs have dependency information, so a user selecting oci_tao at install time will automatically get oci_ace and one of the oci_tao_host_* packages.

The **oci_ace** package contains a VxWorks Layer that (when included in a VxWorks Source Build) builds a number of ACE libraries for VxWorks 7. Aside from the core ACE library (`libACE.a` or `libACE.so`), this package includes ACE's utilities for XML and ETCL parsing.

Building both TAO and applications that use TAO requires the tao_idl compiler (and the ace_gperf utility that it uses). Those programs generate code at build-time so they are not in VxWorks Layers but instead they are separate RPMs (**oci_tao_host_linux** or

**oci_tao_host_windows**) that install binaries to the `partners` directory in the top-level VxWorks 7 installation.

The **oci_tao** package contains a VxWorks Layer that builds TAO's libraries, utilities, and ORB Services. The libraries and executables built include everything in the TAO_ACE workspace (TAO_ACE.mwc) that's applicable to the VxWorks target environment.

Since OpenDDS uses a separate code generator, opendds_idl, in addition to TAO's code generation tools, there are separate host-side OpenDDS packages (**oci_opendds_host_linux** or **oci_opendds_host_windows**).

The **oci_opendds** package contains a VxWorks Layer that builds the OpenDDS middleware libraries along with some executable tools and examples (based on OpenDDS_no_tests.mwc).

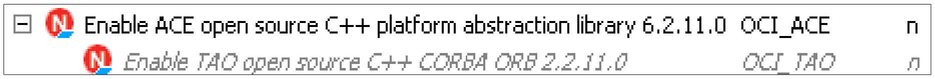
The three packages that install VxWorks Layers use the "net" subdirectory so they can be found at `{WIND_BASE}/pkgs/net/{package}-{version}`.

The version numbers for these RPMs contain the released product version numbers (first three digits) followed by an RPM revision number (final digit). Note that OCI ACE+TAO when treated as a unit takes on TAO's version number. The corresponding ACE release is always exactly 4 major versions ahead. The third digit for OCI ACE and TAO RPMs is the patch level, therefore general release ACE+TAO 2.2a_p12 becomes RPMs 6.2.12._ (ACE) and 2.2.12._ (TAO).

The ACE and TAO RPMs version 6.2.12.0/2.2.12.0 contain minor updates compared to the general release of ACE+TAO 2.2.a_p12. These changes are noted in the files ACE/OCIChangeLog and TAO/OCIChangeLog that are installed along with the source code in their respective layers. The OpenDDS RPM version 3.12.0.0 also contains minor updates compared to the general release of OpenDDS 3.12. These changes are noted in the OpenDDS/ChangeLog file within the installed layer directory.

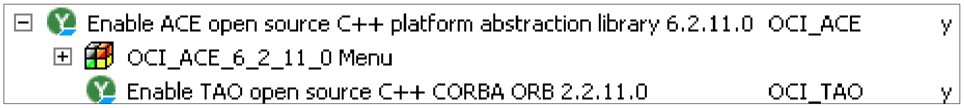
# Configuring a VxWorks Source Build (VSB)

To start using ACE/TAO/OpenDDS, create a new VSB or configure an existing VSB to enable the required layers. See the Wind River VxWorks 7 document "Configuration and Build Guide" for more information on using the VSB Configuration tool in Wind River Workbench (or the command line). The ACE, TAO, and OpenDDS layers start off as excluded. They can be found in the "net" category:



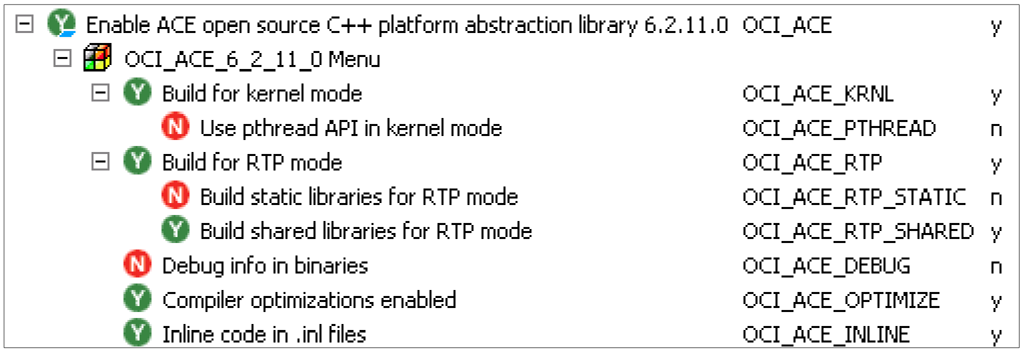
| | | |
|---|---|---|
| ⊟ Enable ACE open source C++ platform abstraction library 6.2.11.0 | OCI_ACE | n |
| Enable TAO open source C++ CORBA ORB 2.2.11.0 | OCI_TAO | n |

Since ACE is not currently included, TAO and OpenDDS (if installed) are showed as greyedout and italicized. Select any layer you'd like to include, right-click it and use the “Add with Dependencies” command from the context menu. In the example below, TAO was added (which automatically added ACE) and OpenDDS was not even installed:



```
⊟ Y Enable ACE open source C++ platform abstraction library 6.2.11.0  OCI_ACE  y
    ⊞ OCI_ACE_6_2_11_0 Menu
    Y Enable TAO open source C++ CORBA ORB 2.2.11.0                   OCI_TAO  y
```

Note that the OCI_ACE layer now has a submenu with a [+] icon to the left. There are additional configuration options available under ACE. Since these options change how ACE is compiled, they are controlled by the ACE layer and are not separately configurable under TAO and OpenDDS (though the settings also impact how TAO and OpenDDS are built).



```
⊟ Y Enable ACE open source C++ platform abstraction library 6.2.11.0  OCI_ACE             y
    ⊟ OCI_ACE_6_2_11_0 Menu
        ⊟ Y Build for kernel mode                                       OCI_ACE_KRNL        y
              N Use pthread API in kernel mode                          OCI_ACE_PTHREAD     n
        ⊟ Y Build for RTP mode                                          OCI_ACE_RTP         y
              N Build static libraries for RTP mode                     OCI_ACE_RTP_STATIC  n
              Y Build shared libraries for RTP mode                     OCI_ACE_RTP_SHARED  y
          N Debug info in binaries                                      OCI_ACE_DEBUG       n
          Y Compiler optimizations enabled                              OCI_ACE_OPTIMIZE    y
          Y Inline code in .inl files                                   OCI_ACE_INLINE      y
```

When the VSB is built, the selected ACE, TAO, and OpenDDS libraries and executables will be built alongside the VxWorks source code. Since there are interactions among the ACE, TAO, and OpenDDS layers, the “Enable Parallel Builds” option in workbench is not supported.

## Building Applications Using ACE, TAO, and OpenDDS

Include files for ACE, TAO, and OpenDDS are installed into the VSB. The C++ preprocessor include path should contain `{VSB_DIR}/krnl/h/public` or `{VSB_DIR}/usr/h/public` for kernel-mode or RTP-mode respectively.

Libraries for ACE, TAO, and OpenDDS are located in the VSB directory in `{VSB_DIR}/krnl/{CPU}/gnu` or `{VSB_DIR}/usr/lib/gnu`.

VxWorks executables for TAO and OpenDDS are located in `{VSB_DIR}/dkm` (Dynamic Kernel Modules) or `{VSB_DIR}/usr/root/gnu/bin` (RTPs). ACE doesn't include any VxWorks executables.

Host executables (code generators) are installed in `{WIND_HOME}/partners/oci_tao-{VERSION}/{HOST_OS}/bin` and `{WIND_HOME}/partners/oci_opendds-{VERSION}/{HOST_OS}/bin`.

# Using MPC to Generate Makefiles for Application Code

MPC: The Makefile, Project, and Workspace Creator (http://objectcomputing.com/products/mpc) is installed inside the oci_ace layer (in the MPC subdirectory). All of the makefiles included in the ACE, TAO, and OpenDDS layers were generated by MPC. Since ACE, TAO, and OpenDDS use MPC to generate their makefiles, the included MPC project base files (*.mpb) contain reusable bits of information that can be used to generate makefiles for your own codebase that uses ACE, TAO, and OpenDDS.

The following example shows how an OpenDDS application can be set up to use MPC with the VSB-based ACE, TAO, and OpenDDS. Since OpenDDS applications also implicitly use ACE and TAO, the example can be adapted for by with ACE-only and TAO-only applications. This example assumes the “wrenv” utility from Wind River has already been run to set up the VxWorks environment. This example is written for a Linux host but it can also be adapted to work on a Windows host.

Set up the environment variables:

| Name | Value |
|---|---|
| VSB_DIR | Directory containing the VxWorks Source Build |
| MPC_ROOT | $WIND_PKGS/net/oci_ace-6.2.12.0/MPC |
| ACE_ROOT | $VSB_DIR/src/oci_ace-6.2.12.0 |
| TAO_ROOT | $VSB_DIR/src/oci_tao-2.2.12.0 |
| DDS_ROOT | $VSB_DIR/src/oci_opendds-3.12.0.0 |
| TAO_HOST_TOOLS | $WIND_HOME/partners/oci_tao-2.2/LINUX |
| OPENDDS_HOST_TOOLS | $WIND_HOME/partners/oci_opendds-3.12/LINUX |

The application's makefile is generated from its .mpc file:

```
project: dcpsexe, dcps_rtps_udp {
  TypeSupport_Files {
    Messenger.idl
  }

  // Add the following line for RTP mode:
  libpaths += $(VSB_DIR)/usr/lib/gnu

  // Or add the following line for Kernel mode:
  libpaths += $(VSB_DIR)/krnl/$(CPU)/gnu
}
```

Run MPC to generate makefiles:

```
$ACE_ROOT/bin/mwc.pl -type gnuace
```

Run make:

Determine additional arguments for “make” based on the ACE settings in the VSB configuration.  If the VSB is configured as shown in the first column, set the makefile variable in the remainder of that row:

| VSB Configuration Setting | Makefile Variable | Value | Notes |
|---|---|---|---|
| Build for kernel mode | rtp | 0 | |
| Use pthread API in kernel mode | pthread | 1 | |
| Build static libraries for RTP mode | static_libs_only | 1 | Also use this for *any* kernel build |
| *NO* - Debug info in binaries | debug | 0 | Defaults to 1, specify 0 if “NO” |
| *NO* - Compiler optimizations enabled | optimize | 0 | Defaults to 1, specify 0 if “NO” |
| *NO* - Inline code in .inl files | inline | 0 | Defaults to 1, specify 0 if “NO” |
| Select the Primary Tool (Wind River) | TOOL | gnu | Specify TOOL=gnu if VSB doesn’t default to it |

For example, if the VSB was set to build for kernel mode, use the pthread API, no debugging info, with optimizations and with inline code:

```
make rtp=0 pthread=1 static_libs_only=1 debug=0 TOOL=gnu
```

To build for RTP mode with the same settings as above except for using shared libraries:

```
make debug=0 TOOL=gnu
```

# Resources and Documentation

- OCI ACE http://theaceorb.com/product/aboutace.html
- OCI TAO http://theaceorb.com/
  - TAO Developer's Guide http://theaceorb.com/purchase/
- OpenDDS http://opendds.org/
  - OpenDDS Developer's Guide http://download.objectcomputing.com/OpenDDS/OpenDDS-latest.pdf
- MPC http://objectcomputing.com/products/mpc
- OCI Commercial Support http://objectcomputing.com/products/open-source-support
- OCI Training https://objectcomputing.com/training/