ADLINK

VORTEX
OPENSPLICE

Migration Guide
Release 6.x

Vortex OpenSplice

MIGRATION GUIDE

Copyright Notice

Preface

About this Migration Guide

This *Migration Guide* is intended to help users migrate their existing code from OpenSplice version *5.x* to OpenSplice version *6.x*.

⚠ This guide is **only** intended for customers who are currently using *OpenSplice V5.x* and who want to migrate to *OpenSplice 6.x*. Migration from older versions up to V5.x is covered by separate Migration Guides.

⚠ This guide *does not* cover all of the differences which exist between the OpenSplice version 6.x and previous versions, only those which affect compatibility. Please refer to the Release Notes for a complete list of the changes in the new version.

i The C language binding is provided with a special *legacy mode* which enables pre-version 6.x code to be used without modification.

Conventions

The conventions listed below are used to guide and assist the reader in understanding the Migration Guide.

⚠ Item of special significance or where caution needs to be taken.

i Item contains helpful hint or special information.

WIN Information applies to Windows (*e.g.* XP, 2003, Windows 7) only.

UNIX Information applies to Unix based systems (*e.g.* Solaris) only.

C C language specific.

C++ C++ language specific.

Java Java language specific.

Hypertext links are shown as *blue italic underlined*.

On-Line (PDF) versions of this document: Items shown as cross references, *e.g. Contacts* on page iv, act as hypertext links: click on the reference to go to the item.

```
%  Commands or input which the user enters on the
   command line of their computer terminal
```

Courier fonts indicate programming code and file names.

Extended code fragments are shown in shaded boxes:

```
NameComponent newName[] = new NameComponent[1];

// set id field to "example" and kind field to an empty string
newName[0] = new NameComponent ("example", "");
```

Italic and ***Italic Bold*** indicate new terms, or emphasise an item.

Sans-serif and **sans-serif bold** indicate Graphical User Interface (GUI) and Integrated Development Environment (IDE) elements and commands; for example, 'click the Cancel button' and 'choose **File > Save**'.

Step 1: One of several steps required to complete a task.

Contacts

ADLINK can be reached at the following contact points for information and technical support.

USA Corporate Headquarters	European Head Office
ADLINK Technology Corporation	ADLINK Technology Limited
400 TradeCenter	The Edge
Suite 5900	5th Avenue, Team Valley
Woburn, MA	Gateshead
01801	NE11 0XA
USA	UK
Tel: +1 781 569 5819	Tel: +44 (0)191 497 9900

Web: *http://ist.adlinktech.com*
E-mail: *ist_info@adlinktech.com*

ADLINK

MIGRATION TO VERSION 6.X

1 *Critical differences*

This guide is intended to help users of OpenSplice Version 5 with the migration of their existing code-base to the new version of OpenSplice (V6.x).

Not all changes in the product are described here, as this document is focused on those changes that can cause incompatibilities between existing code and the new version of OpenSplice.

A complete list of all changes in the product can be found in the Release Notes.

1.1 OpenSplice configuration file

In OpenSplice V6.x a new *required* configuration element called `Id` is added to the `Domain` tag.

This element specifies the domain Id of the instantiated DDS domain. If several different DDS domains are required to run simultaneously, then they must all have their own unique domain Id.

Example `config` file for OpenSplice V6.x (the **bold** element needs to be added):

```
<Domain>
      <Name>OpenSpliceV6</Name>
      <Id>1</Id>
      <Database>
         <Size>10485760</Size>
      </Database>
      <Lease>
         <ExpiryTime update_factor="0.05">60.0</ExpiryTime>
      </Lease>
</Domain>
```

This gives the domain an Id value of **1**.

1.2 API call changes

In OpenSplice V6.x the `create_participant`, `lookup_participant`, `lookup_domain` and `get_domain_id` API calls are changed.

The QoS settings *ReaderLifecycle* and *DataWriter reliability* have also changed.

1.2.1 Parameter *DomainId_t* is now an integer

⚠️ The parameter `DomainId_t` is now changed from a *string* to an ***integer***. The new constant `DDS_DOMAIN_ID_DEFAULT` can be used for this parameter. If this is done the value of new Id tag from the configuration file specified by the environment variable called `OSPL_URI` will be used.

It is recommended that you use this domain Id in conjunction with the `OSPL_URI` environment variable instead of hard-coding a domain Id into your application, since this gives you much more flexibility in the deployment phase of your product.

The example code below illustrates this usage (differences indicated in **bold**).

Example using OpenSplice V5 with the SAC API

```
DDS_DomainParticipantFactory dpf;
DDS_DomainParticipant dp;
DDS_DomainId_t domain = "";
dpf = DDS_DomainParticipantFactory_get_instance();
/* Create a domain participant with a NIL listener and
 * "" as the value of the DomainId_t parameter.
 */
dp = DDS_DomainParticipantFactory_create_participant (
    dpf,
    domain,
    DDS_PARTICIPANT_QOS_DEFAULT,
    NULL,/* NIL listener */
    DDS_STATUS_MASK_NONE);
```

Example using OpenSplice V6.x with the SAC API

```
DDS_DomainParticipantFactory dpf;
DDS_DomainParticipant dp;
DDS_DomainId_t domain = DDS_DOMAIN_ID_DEFAULT;
dpf = DDS_DomainParticipantFactory_get_instance();
/* Create a domain participant with a NIL listener and
 * DDS_DOMAIN_ID_DEFAULT as the value of the DomainId_t parameter.
 */
dp = DDS_DomainParticipantFactory_create_participant (
    dpf,
    domain,
    DDS_PARTICIPANT_QOS_DEFAULT,
    NULL,/* NIL listener */
    DDS_STATUS_MASK_NONE);
```

1.2.2 Change in *ReaderLifecycle* QoS (invalid sample setting)

The QoS setting that determines whether OpenSplice creates invalid samples to communicate state changes, called `enable_invalid_samples`, is now deprecated. Invalid samples, in the past, could only be either `enabled` or `disabled`. There is a replacement QoS setting, called ***invalid_sample_visibility***, which accepts *three* values:

ADLINK

MINIMUM_INVALID_SAMPLES:
Acts like the old `enable_invalid_samples = true`, an invalid sample will be created if there's no regular sample on which the state change can be piggy-backed (this is the default behavior).

NO_INVALID_SAMPLES:
Acts like the old `enable_invalid_samples = false`, no invalid samples are created.

ALL_INVALID_SAMPLES:
Currently not implemented, but in the future will create an invalid samples for all state changes even if a regular sample is available.

Using the deprecated setting will cause a warning to be logged to the info log. It will be removed from the product in the future. If applicable it is recommended to switch application code to the new *invalid_sample_visibility* setting instead.

1.2.3 Default *DataWriter reliability* QoS is now set to **RELIABLE**

In the OpenSplice v5 series, the default DataWriter reliability QoS was set to `BEST_EFFORT`. The OMG DDS Specification states that the default value should be **RELIABLE**. For the OpenSplice v6 series this has been corrected.

1.3 OpenSplice license management and environment changes

OpenSplice V6.x uses Reprise License Manager instead of FlexLM to manage licenses. The licensing variables that need to be used have therefore changed.

Replace

PTECH_LICENSE_FILE *with* **prismtech_LICENSE**

LM_LICENSE_FILE *with* **RLM_LICENSE**

You will need a new license file for V6.x. Please contact *ist_info@adlinktech.com* to request one, if you have not already been sent it.

Refer to the chapter on *Licensing OpenSplice* in the *Getting Started Guide* for full details.