IBM Platform Symphony 6.1.1 ICC 11 Support on SLES 10 Readme File

About ICC 11 support on SLES 10
This readme file describes how to compile an application source file with ICC 11 on SLES 10. Use this readme file when you want to develop an application for Platform Symphony 6.1.1 with ICC 11 on SLES 10.

Readme file for: IBM® Platform Symphony
Product/Component Release: Symphony 6.1.1
Update Name: Fix pack
Fix ID: sym-6.1.1-build228818-bbva
Publication date: 09 Jan 2014
Last modified date: 09 Jan 2014

1. Scope

Applicability	
Operating system	SUSE LINUX ENTERPRISE SERVER 10 WITH SP2 X64
Symphony version	6.1.1
Cluster types	This feature applies to a single-cluster grid, DE cluster, and Client.
Dependencies	
File system	N/A
Limitations	
ICC Version	11.1
Security Support	No extension of security for Symphony, such as for GSS-Kerberos security plug-in.
Known Issues	N/A

2. Configuration to enable ICC 11 Support on SLES 10

No package needs to be installed, and no Symphony configuration is required to enable this feature.

1) Prerequisites

- Symphony 6.1.1 must be installed.
- ICC 11 must be installed on development hosts.

3. Usage

1) How this feature works

With the installation of ICC, you can use “icc” instead of “gcc/g++” to build your application on development hosts. In other words, you need to:

- Rebuild your CPP client and service with ICC 11.
- Deploy/update the new service binaries in the cluster.

2) Rebuild with ICC 11

If you build your application with commands, replace “g++/gcc” with “icc” in your compiling command, for example:

```
> icc -c -o sampleApp.o sampleApp.cpp
```

If you build your application with makefile, edit the makefile by commenting out the “CC” environment with “g++/gcc” and adding the “CC” environment with “icc” as follows:

```
# add the following line in Makefile
CC = icc
# comment out the following line in Makefile
#CC = g++
```

3) Deploy/update the service package

For a new application, you must deploy the service package into the cluster. You can refer to the following section in the “Knowledge Center” for detailed steps of “deploying a new application”:

```
Develop -> Application Development Guide -> Application Deployment and
Management -> Service Package Deployment -> Deploying a new application
```

For an existing application, you must update the service package on the cluster. You can refer to the following section in the “Knowledge Center” for detailed steps of “updating an existing application”:

```
Develop -> Application Development Guide -> Application Deployment and
Management -> Updating applications -> Change a service package for an
existing service/Change a service package using the CLI
```

4) Example

SharingData under “${SOAM_HOME}/6.1.1/samples/CPP/SharingData” is an application sample of Symphony DE. You can refer to the “Application Samples” section in the “DE Knowledge Center” for more information about application management.

1. Rebuild the client and service by ICC 11 on the development host:
   Update the three makefiles under the directories of “Common”, “Service”, and “Client” as follows:

   ```
   # add the following line in Makefile
   CC = icc
   # comment out the following line in Makefile
   # CC = g++ $(BIT64_FLAG) -Wno-deprecated -Wall
   ```

   Execute the “make” command under the top directory of “SharingData”. New binaries can be found in the “Output” directory.

2. Deploy/update the new service package in the Symphony cluster:
   Compress a new service package with the newly built binary under the “Output” directory.

   ```
   > tar zcvf SharingDataServiceCPP.tar.gz SharingDataServiceCPP
   ```

   Deploy/update the new service package in the cluster:

   ```
   > soamdeploy add SharingDataServiceCPP -p SharingDataServiceCPP.tar.gz
     -c /SharingDataCPP
   ```

   If the application has not been registered, execute the following command under the top directory of “SharingData”:

   ```
   > soamreg SharingData.xml
   ```

3. Run the newly built client on client hosts in the Symphony cluster:
   Execute the newly built client binary on the client hosts; the application runs successfully:

   ```
   > ./DataClient
   ```

4. Copyright and trademark information