OPC Toolbox Update Discription Version 3.04 (07/15/2001)

OPC C++ Toolkits

SOCmn (common library):

- The method pop of class SOCmnStringList returns now the first string of the list.
- The Object-Log Trace in the destructor of SOCmnTrace are now encapsulated by try-catch blocks to provide for exceptions.
- The SOCmnString class was enhanced by the makeUpper method.
- An error during searching elements of an SOCmnStringList was eliminated.
- The SOCmnVariant class now additionally supports the VARIANT data types VT_CY and VT_CY | VT_ARRAY.

SOSrv (server-specific library):

- An error during life cycle control of the object references to the root object of the DA name space and the AE area space was eliminated. This problem occurred when a string-based node was used directly below the root.
- As of version 3.04, the Toolkit also supports the Swedish, Russian and Hungarian languages in addition to English and German which were already available before.
- The SOSrvEntry class was enhanced by the initializeSecurity method. This method enables the server application to indicate the authentication and impersonation levels which are demanded by the cClient and offered to it during notifications.
- For some success error codes (e.g. OPC_S_INVALIDMAXSIZE), the OPC interface function IOPCCommon::GetErrorString did not supply the OPC error text but the text of the system error code. Now the OPC error text is supplied.
- Up to version 3.04, calling of the method SOSrvEntry::shutdownAllClients could lead to a deadlock with OPC Clients if these madke COM calls() to the server in the IOPCShutdown::ShutdownRequest function.
- In version 3.04, the Server Toolkits allow for the implementation of several instances of an OPC DA or AE Server in an OPC Server Binary. An instance of an OPC Server is characterized by a separate CLSID and a separate configuration. For the implementation of this extension, an additional parameter was introduced in the methods SODaSCreator::createServer and SOAeSCreator::createServer. If you have overloaded these methods in a server application, they must be adapted in this application! In the methods SODaSEntry::getNameSpaceRoot, SOAeSEntry::getAreaSpaceRoot, SOAeSEvent::fire and SOAeS::fireEvents, an additional optional parameter for the server instance is introduced. The classes SOSrvServer and SOAeSCondition contain an additional method, getServerInstance, for querying the server instance. With the method addComObjMapEntry of the classes SODaSEntry, SOAeSEntry and SOAsSEntry, the instances of the OPC Server can be set up. Samples for demonstrating this new functionality can be found in the directories samples/SODaS/MulitServerDLL, samples/SOAeS/MulitServerEXE and samples/SOAsS/MulitServerDLL.

SODaS (OPC Data Access Server library):

- In Version 3.00 an asynchronous Refresh from OPC clients did not deliver values, if the Refresh was performed for ItemTags, that uses report UpdateMode.
- A new sample for using the string based name-space was added.
- The OPC Data Access Server Library now statically integrates the registry functionality of the ATL.
- In vVersion 3.04, it is possible to indicate EU information for items by means of which the value of a property is accessed. For this purpose, the methods getEUInfo and getEUType awere moved from the SODaSTag class to theirits basic class SODaSItemTag and are now virtual.
- The OPC interface function IOPCItemMgt::SetActiveState now supplies an error if the activation of an item by the method SODaSItem::setActive has failed.
- The OPC interface functions IOPCItemProperties::LookupItemIDs and IOPCItemProperties::GetItemProperties now supply S_FALSE if the action has failed for at least one property has failed. This return value for the functions mentioned above is now described in the OPC Data Access Specification V2.04.

- For the transmission of data modification messages and the finishing of I/O requests, additional checks for NULL pointers() were integrated.
- A conversion of data is now also possible in dependence on the ItemTag object with the corresponding data. For this purpose, an optional parameter was added to the method SODaSCache::changeDatatype.
- VT_CY and VT_CY | VT_ARRAY were added to the possible variant data types of an item.
- The property in the ID 108 (Item Timezone) was removed from the standard properties for node and tag objects.
- Up to version 3.04, the 'fully qualified item id' of a property was fixed to the following scheme: #. Now the part after the property delimiter (#) can be freely determined by the application. For this purpose, the following methods were changed: SODaSCreator::createProperty, SODaSServer::isPropertyName, SODaSServer::createProperty, SODaSTag::addProperty. The following methods were implemented in addition: SODaSServer::getPropertyIDByName, SODaSServer::getPropertyItemID.
- Node objects created dynamically by the Ttoolkit from the string-based name space are now automatically released even if they are not used by an OPC Client anymore. For better initialization of dynamically created objects, the methods SODaSServer::initTag and SODaSServer::initProperty were introduced.
- Access to the Entry object is now synchronized.
- With application-specific server or node objects whose child lists are implemented as maps, there have been problems with the list functions update and findObject.
- The list type parameters in the constructors of the classes SODaSServer and SODaSGroup were removed.
- If items are deleted by the OPC Client (IOPCItemMgt::RemoveItems), all I/O requests not yet processed for these items are now processed with the errors OPC_E_INVALIDHANDLE.
- The performance of the method SODaSGroup::getItemByHandle was improved. The default implementation of the method SODaSGroup::trustClientItemHandles now supplies TRUE.
- For groups with an update rate of 0 ms which contains only 'report-updated' items, the Vversion 3.04 now guarantees that no data modification signalled to the Toolkit is lost. For this purpose, the call parameters of the method SODaSItemTagIConnect::notifyAllConnectedObject awere extended and the methods SODaSGroup::isNotifyAllChanges, SODaSRequest::.setReportValue, SODaSRequest::setReportValue, SODaSRequest::getReportValue, SODaSRequest::setReportQuality, SODaSRequest::getReportQuality, SODaSRequest::.setReportTimeStamp and SODaSRequest::getReportTimeStamp were created. A sample for demonstrating this new functionality can be found under samples/SODaS/everyDataChange.
- In case of write-only tags, reading the values of the OPC standard properties 2 (item value), 3 (item quality) and 4 (item timestamp) via the OPC interface function IOPCItemProperties::GetItemProperties now supplies the error OPC_E_BADRIGHTS.

SOAeS (OPC Alarms and Events Server library):

- The implementation of the QuerySourceConditions method has contained bugs by using string-based conditions.
- After the IOPCEventSink::OnEvent calback has been blocked by a miss-behaving client, events have not been sent any longer to other clients too.
- A new sample for using the string based area-space was added.
- The OPC Alarms and Events Server library now statically integrates the registry functionality of the ATL.
- With version 3.04, condition objects of the class SOAeSCondition can be used which have no pointer to a source object of the area tree. For this purpose, the virtual method getSourceName was introduced in the SOAeSCondition class.
- The parameter dwCookie of a condition-related event notification can now be determined by the server application. It is queried with the virtual method getCookie of the SOAeSCondition class.
- Access to the entry object is now synchronized.
- The methods activate, changeQuality and changeState of the SOAeSCondition class send no notifications to the client anymore if the condition is "disabled".
- The list type parameters in the constructors of the SOAeSServer class were removed.
- The actor id is now set appropriately by the SOAeSCondition class during generation of an event.

- All objects of the SOAeSSubscription class now set the flag SOAESSUBSCRIPTION_FLAG_FORCE_ASYNC_SEND_EVENTS in the constructor. Thus, all events from the event processing thread of the Toolkit are sent to the OPC Clients by default.