

SLIK-DA Control

(Simple Language Independent Kit for Creating OPC Data Access Servers)
 "Quality OPC Server Implementation Solutions for Software Developers"



Contents

OPC AND SLIK-DA 2
HOW SLIK-DA WORKS 3
TUTORIAL—DEVELOP A SIMPLE SERVER 4
PRODUCT SUMMARY SHEET 10



Introduction

The Simple Language Independent Kit for creating OPC Data Access Servers (or SLIK-DA) provides a level of simplicity and flexibility not available before in an OPC server software development toolkit.

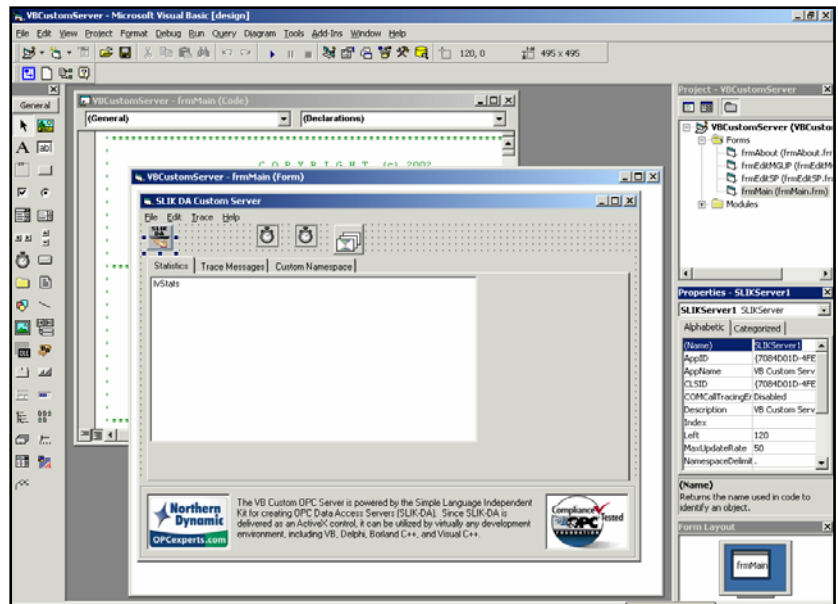
SLIK-DA is an ActiveX and .NET Windows Form Control used to quickly add OPC server DA interfaces to your C#, VB.NET, or Visual Basic application.

Just define 6 properties, call 4 methods, implement 2 event handlers, and provide an update method for your server's tags. It can be as simple as that!

This tutorial document will demonstrate to you just how easy SLIK-DA is to use.

The instructions in this tutorial document step you through the creation of a simple server that simulates device or system points.

For a more comprehensive examination of server source code, download and install our free evaluation version of the SLIK-DA Control. The evaluation software includes documented C#, VB.NET, VB, and VC++ source code for sample servers.



Free Evaluation Software

A free 30 day full-featured evaluation version of SLIK-DA is available to download from our website. The URL is <http://www.OPCexperts.com/SLIKDA>. Then click on Free Evaluation. You can build your own OPC server with the evaluation software.

OPC & SLIK-DA

OPC (or OLE for Process Control) is a set of publicly available technical standards for enabling the interoperation of factory automation software applications. The OPC standards are developed by factory automation equipment suppliers and end users, through an organization called the OPC Foundation.

Presently, OPC standards exist for the following types of factory automation software interoperation:

- Data Access (DA) – real time data
- Alarms & Events (A&E) – real time alarm / event data
- Historical Data Access (HDA) – historian data
- Batch –batch data

The OPC standards require software applications to adopt a role for a data interchange. Two roles are defined, a server and a client. A server is a producer of data in the exchange. Examples include servers for PLCs and DCSs. SLIK-DA is used by software developers and system integrators to implement DA servers. A client is a consumer of data in the exchange. Examples include SCADA/HMI software and Northern Dynamic's OPC Gateway.

OPC Data Access Server Overview

An OPC DA client connects to an OPC DA server and manages the data interchange using a number of objects, specifically the server, group, and item objects.

Server Object – This object represents the server to which

a client has requested a connection. The object is also a container for all the group objects created by the client.

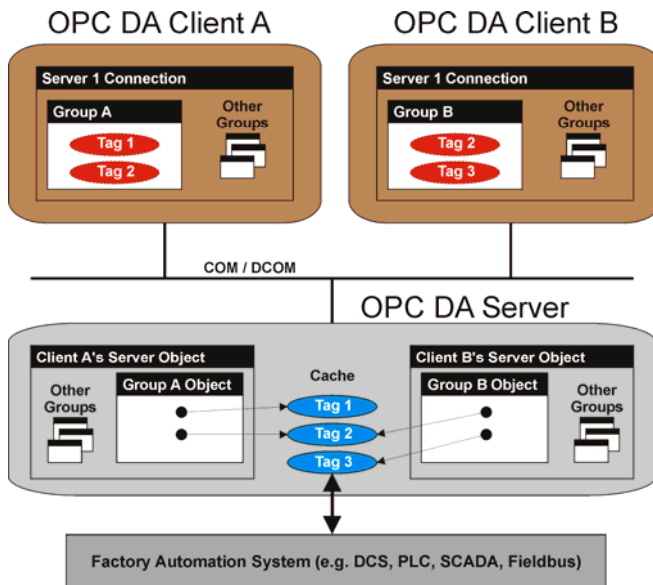
Group Object – This object is a container for organizing one or more items. Clients create and manage groups in a manner that suits their application.

Item Object – Items are the real-time data references. Servers present a list of items that are available to a client. The client chooses which items to use, how to organize them and how to use them. A client can choose to read, write or receive unsolicited updates for an item. Each item has three values associated with it: a value, a timestamp, and a quality indicator.

SLIK-DA Benefits

Why use the SLIK-DA Control to develop your OPC DA server?

- Reduce your development time.
- Focus your implementation efforts on the high value adding features of your server application.
- SLIK-DA hides the intricacies of COM interface definitions and the OPC DA standard.
- Lower your server's lifecycle maintenance costs. Future SLIK-DA updates implementing OPC specification changes are easy to apply.
- Lower integration costs by relying on our OPC expert support and SLIK-DA's built-in diagnostic tools.
- Leverage the experience and expertise of Northern Dynamic OPC developers.



This diagram illustrates a logical view of the run-time interaction between an OPC DA server and two OPC DA clients. Servers can simultaneously manage connections with one or more clients. While clients create groups and populate the groups with item references, servers manage the groups and item references. Typically, items are managed by a server as a cache or real-time database. How the cache is monitored and updated is server implementation specific. How the clients are updated is specified by the OPC DA standard.

How SLIK-DA Works

Overviews

SLIK-DA is an ActiveX control that you use in your OLE automation enabled development environment, such as Visual Basic and Delphi, to create OPC Data Access servers.

SLIK-DA provides a programming model that will be familiar to Visual Basic and ActiveX control users.

The diagram shown below illustrates how SLIK-DA ActiveX is used in your OPC server application. The OPC DA server objects (OPCServer, OPCGroup, OPC Item) are automatically created and managed by SLIK-DA. No knowledge of OPC DA or COM is required. You write application code that:

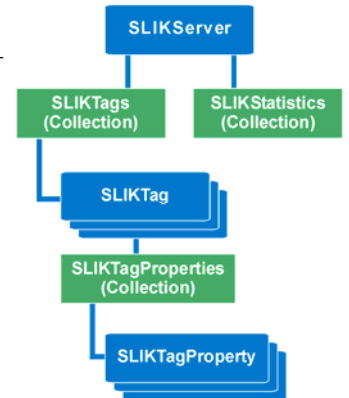
1. **Initializes SLIK-DA,**
2. **Provides event handlers for the device read and write operations, and**
3. **Provides a mechanism for updating the SLIKTag in the SLIKTags collection.**

SLIK-DA takes care of the rest!

Programming Model

The programmer works with a simple object hierarchy, modifying properties, calling methods, and implementing event handlers. The diagram on the right shows the objects in SLIK-DA's programming model.

SLIKServer. This object is the primary interface to the SLIK-DA ActiveX Control. SLIKServer provides properties and methods used to make your server unique. The object also maintains a collection of all the available tags (SLIKTags) and pre-defined server statistics (SLIKStatistics).

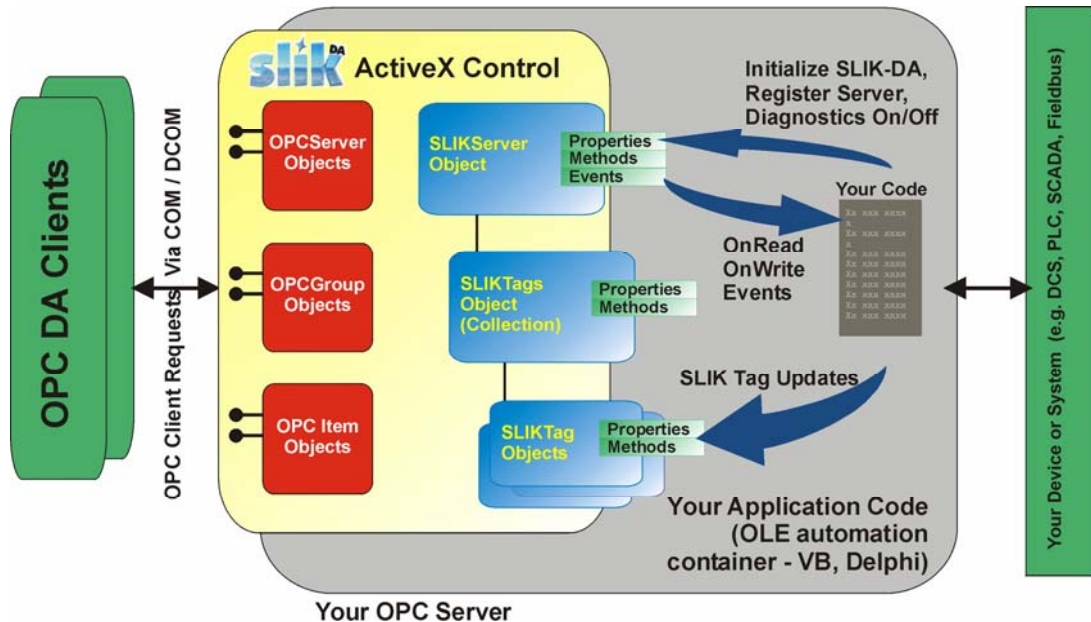


SLIKTags (Collection).

Contains all the current tags (SLIKTag objects) in an OPC server. Each tag is uniquely identified by a fully qualified name. SLIK-DA uses this collection to implement browsing, a feature that allows OPC clients to view items available from your server.

SLIKTag. An object that represents a unique data point in your OPC server. Each SLIKTag object corresponds to an OPC item that a client can access from your server. This item can be read or written by an OPC client, depending on how you define the SLIKTag object.

SLIKTagProperty. A SLIKTag can have one or more properties associated with it. OPC clients can read these properties. Use this object (via the SLIKTagProperties collection) to define engineering unit maximum and minimum values, for example.



Tutorial—Develop a Simple Server

Pre-Requisites

This tutorial assumes that you have the following software installed.

SLIK-DA Control evaluation software available for download from:

<http://www.OPCexperts.com/SLIKDA>

Microsoft Visual Basic 6.0.

Tutorial Overview



The instructions in the following Tutorial Steps section illustrate how quickly you can develop an OPC framework for your own server using SLIK-DA ActiveX and Visual Basic.

Follow the steps to create a simple OPC Data Access server that simulates tags/points/items from an automation system.

As an overview, here is what you will be doing.

1. **Define the parameters that make your server unique to world of OPC clients.**
2. **Create code that processes an OPC client device read request. SLIK-DA will tell your application code what to read and how often. (OPC clients can request either cache and device reads. SLIK-DA services the cache reads for our application.)**
3. **Create code that will process an OPC client's request for a device write. SLIK-DA will tell your application when and what to write.**
4. **Create code that will keep your SLIKTag cache up to date with data from your underlying system. Here you have several design options. The option you choose will depend on how your underlying system (API, protocol stack, etc.) works.**

Tutorial Steps

1. **Start Visual Basic and create a new Standard EXE project. Add the SLIK-DA ActiveX control to your project's toolbox.**
 - [Project->Components](#)
 - Select [Northern Dynamic's SLIK-DA ActiveX Control](#) from the Controls list and click OK.
 2. **Create an instance of the control for your project.**
 - Select the SLIK-DA icon  in the toolbox.
 - Drag the cursor on the default form to create an instance of SLIK-DA in your project. You will see the following icon on your default form.
- 
3. **Assign a class identifier and an application identifier for your OPC server.**
 - Select the SLIK-DA control on your form.
 - Select the [CLSID](#) property in the VB Properties Window.
 - Click the drop-down arrow on the combo box and select the option [Auto Generate](#).
 4. **Assign an application identifier for your OPC server.**
 - Select the SLIK-DA control on your form.
 - Select the [AppID](#) property in the VB Properties Window.
 - Click the drop-down arrow on the combo box and select the option [Auto Generate](#).

NOTE : The Visual Basic code presented in steps 5, 6, 7, and 8 are to be added to the Load event handler for the form where the SLIK-DA control was placed, in the order shown.

(Continued on page 5)

LISTING A - Define the parameters that make your server unique to an OPC client

```
SLIKServer1.ProgId = "NDI.MySLIKDADemo.1"
SLIKServer1.AppName = "My SLIK-DA Demo Server"
SLIKServer1.Description = "Demo server built using SLIK-DA ActiveX"
SLIKServer1.VendorName = "NDI"
```

Tutorial—Develop a Simple Server

(Continued from page 4)

5. **Define the parameters that describe your server to OPC clients when they browse for available OPC servers.**
 - Set the following SLIKServer object properties to uniquely define your OPC server:
 - ProgID: A version dependent programmatic identifier for your server,
 - AppName: The name of your application,
 - Description: A brief description of your server,
 - VendorName: The name of your company.
 - Add the code shown in Listing A to the Load event handler for your form. NOTE: be sure to include the version number (.1) for ProgID!
6. **Create the application code that handles the regserver and unregserver command line arguments.**
 - Add the code shown in Listing B to the Load event handler for your form.
7. **Create the collection of tags your server will make available to an OPC client.**
 - Add the code shown in Listing C to create four read/write tags in one of two namespace folders.
 - The sample code creates one integer tag, one 4 byte real tag, one string tag, and one Boolean tag.
8. **Start the server. Now OPC clients can connect to your server using the OPC Data Access interfaces provided by SLIK-DA.**
 - Add the following SLIKServer object method call to your form's Load event handler:
 - SLIKServer1.StartServer

We are done modifying the form's Load event handler.
9. **Define an OnRead event handler for the SLIK-DA's SLIKServer object. The OnRead event handler is used to process OPC client device read requests. The implementation of this event handler is mandatory.**
 - In the VB Code Window, select SLIKServer1 from the Object Box.
 - Now select OnRead from the Procedures/ Events Box.
 - The Sub procedure SLIKServer1_OnRead is created. Here is where you add the code necessary to read the value, quality and timestamp from your system or device. This may be an API call, memory access, etc.
 - For demonstration purposes, the sample code

(Continued on page 6)

LISTING B – Use helper methods to process your server's command line arguments

```
Dim sCmdLine as String
sCmdLine = Command()
If InStr (1, sCmdLine, "unregserver", vbTextCompare) > 0 Then
    SLIKServer1.UnregisterServer
    Unload Me
    Exit Sub
ElseIf InStr (1, sCmdLine, "regserver", vbTextCompare) > 0 Then
    SLIKServer1.RegisterServer
    Unload Me
    Exit Sub
End If
```

Listing C – Create a static tag collection (or cache) for your server

```
Dim Tags as ISLIKTags
Dim Tag as ISLIKTag
Dim nAccess as Long

Set Tags = SLIKServer1.SLIKTags
nAccess = sdaReadAccess + sdaWriteAccess
Set Tag = Tags.Add( "Folder 1.Integer Item", nAccess, Cint(10), sdaGood)
Set Tag = Tags.Add( "Folder 1.Real Item", nAccess, CSng(1000), sdaGood)
Set Tag = Tags.Add( "Folder 2.String Item", nAccess, CStr("Hello World"), sdaGood)
Set Tag = Tags.Add( "Folder 2.Boolean Item", nAccess, True, sdaGood)
```

Tutorial—Develop a Simple Server

(Continued from page 5)

shown in Listing D just increments the value of the integer item.

10. Define an OnWrite event handler for the SLIK-DA's SLIKServer object. The OnWrite event handler is used to process OPC client write requests. The implementation of this event handler is mandatory.

- In the VB Code Window, select SLIKServer1 from the Object Box.
- Now select OnWrite from the Procedures/Events Box.
- The Sub procedure SLIK_Server1_OnWrite is created. Add the code necessary to write the value to your system or device. This may be an API call, memory access, etc.
- For demonstration purposes, we will assume values written to the 4 byte real item are successful and all other write requests fail. Refer to Listing E.

11. The final step to creating a server is to update the Tags you have created (i.e. "Folder 1.Integer Item", "Folder 1.Real Item", etc.) with data from your

system or device. There are two general design approaches for updating the SLIKTag objects.

- SLIK-DA, using the optional *OnUpdate* event handler, signals your code to acquire the data.
- Your application acquires the data on it's own and updates the SLIKTag objects as the data is received by your application code.

NOTE: The section **Updating Tags in the SLIKTags Collection** on page 8 discusses these two approaches in more detail. For the purposes of this tutorial, the addition of this code is left as an exercise for the reader.

12. Build the server executable image.

- File-> Make Project1.exe (or your project name)

13. Register your OPC server as an OPC compliant COM object on your computer. From the command line, run your OPC server specifying the regserver switch.

- Project1.exe -regserver

(Continued on page 7)

Listing D – Sample code for you server's OnRead event handler

```
Private Sub SLIKServer1_OnRead(ByVal Count As Long, Tags() as NDISLIKAct1.ISLIKTag, Access-
Paths() as String, Errors() as Long, Result as Long)

Dim nIndex as Integer
Dim Tag as ISLIKTag
For nIndex = 0 to (Count - 1)
    Set Tag = Tags(nIndex)
    If (Tag.Name = "Folder 1.Integer Item") Then
        Call Tag.SetVQT((Tag.Value + 1), sdaGood, Now())
    End If
Next nIndex
End Sub
```

Listing E – Sample code for you server's OnWrite event handler

```
Private Sub SLIKServer1_OnWrite(ByVal Count as Long, Tags() as NDISLIKAct1.ISLIKTag, Access-
Paths() as String, Values() as Variant, Errors() as Long, Result as Long)

Dim nIndex as Integer
Dim Tag as ISLIKTag
For nIndex = 0 to (Count - 1)
    Set Tag = Tags(nIndex)
    If (Tag.Name = "Folder 1. Real Item") Then
        Call Tag.SetVQT(Values(nIndex), sdaGood, Now())
        Errors(nIndex) = sdaOK
    Else
        Errors(nIndex) = sdaEFail
    End If
Next nIndex
Result = sdaOK
End Sub
```

Tutorial—Develop a Simple Server

(Continued from page 6)

NOTE: When you are done experimenting with your demo server, be sure to unregister your server to keep your registry clean. Use the following command:

- Project.exe -unregserver**

Now you are ready to connect to the server using your OPC client application. This completes the tutorial steps.

If you have any questions about using SLIK-DA ActiveX in your application, please contact our OPC Expert technical support. In addition, the SLIK-DA software installation includes Visual Basic source code for two server implementations, a Simple Server and a Custom Server. The source code is located under the installation sub-directory "Samples\VB6".

Simple Server — Demonstrates SLIK-DA features such as the use of a static tag collection. The server uses the default implementations for browse interface support and OPC item validation.

Custom Server — Demonstrates the flexibility of SLIK-DA. The Custom Server implements a tag collection that is dynamically created as OPC clients make AddItem requests. This server also implements its own browse interface support and OPC item validation. A screen shot from this server is shown below.

Implementation Options

SLIK-DA ActiveX is designed to be simple and flexible. The tutorial steps illustrate how simple it is to create a basic OPC server. SLIK-DA also provides features that allows flexibility in how a server is designed.

Here is a brief list of some of the implementation options available. Refer to the Reference Manual, installed with the SLIK-DA evaluation software, for more information.

Optimize communications with your device by monitoring for Tag active / inactive state changes using the SLIKServer object's OnTagActiveStateChange event.

Rather than defining a static tag collection for your server (as we did in the tutorial), dynamically add and delete SLIKTags from your SLIKTags collection based on OPC client AddItem / RemoveItem requests.

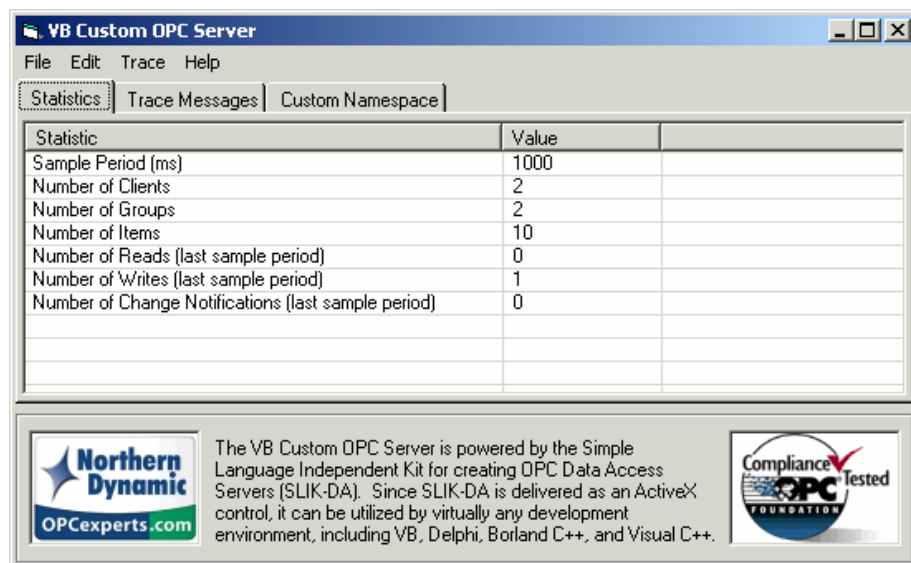
Override SLIK-DA's default OPC item validation to implement custom item validation logic.

Manage client connections using the SLIKServer's OnClientConnect event to limit connections or implement software licensing.

Override SLIK-DA's built-in OPC item browsing to implement your own OPC item browsing.

Integrate SLIK-DA's Three-Tiered Diagnostic Support features with your application's diagnostic tools.

(Continued on page 8)



Tutorial—Develop a Simple Server

(Continued from page 7)

Updating Tags in the SLIKTags Collection

Typically, the majority of OPC interactions between an OPC client and an OPC server are OPC item cache read requests. Cache read requests are very efficient as they are serviced with data stored in your server's memory. These requests can be synchronous (completed while the client waits) or asynchronous (the client is notified when the read completes).

SLIK-DA manages all the OPC-related cache read requests for your application. Your application code is responsible for keeping the in-memory OPC items (or SLIK-Tag objects in the SLIKTags collection) updated with the latest values, quality, and timestamps from your automation system or device.

NOTE: SLIK-DA optimizes the cache so that only one instance of an OPC item (or one SLIKTag) is stored in memory for all OPC client requests. Therefore, your application code is responsible for updating a single OPC item (or SLIK-DA Tag), regardless of the number of clients referencing the item.

SLIK-DA was designed to provide flexibility in how you

choose to keep your SLIKTag objects updated.

There are two design approaches available for implementing your server's SLIKTags update. The approaches differ based on what initiates the SLIKTags update, the SLIK-DA or your application code.

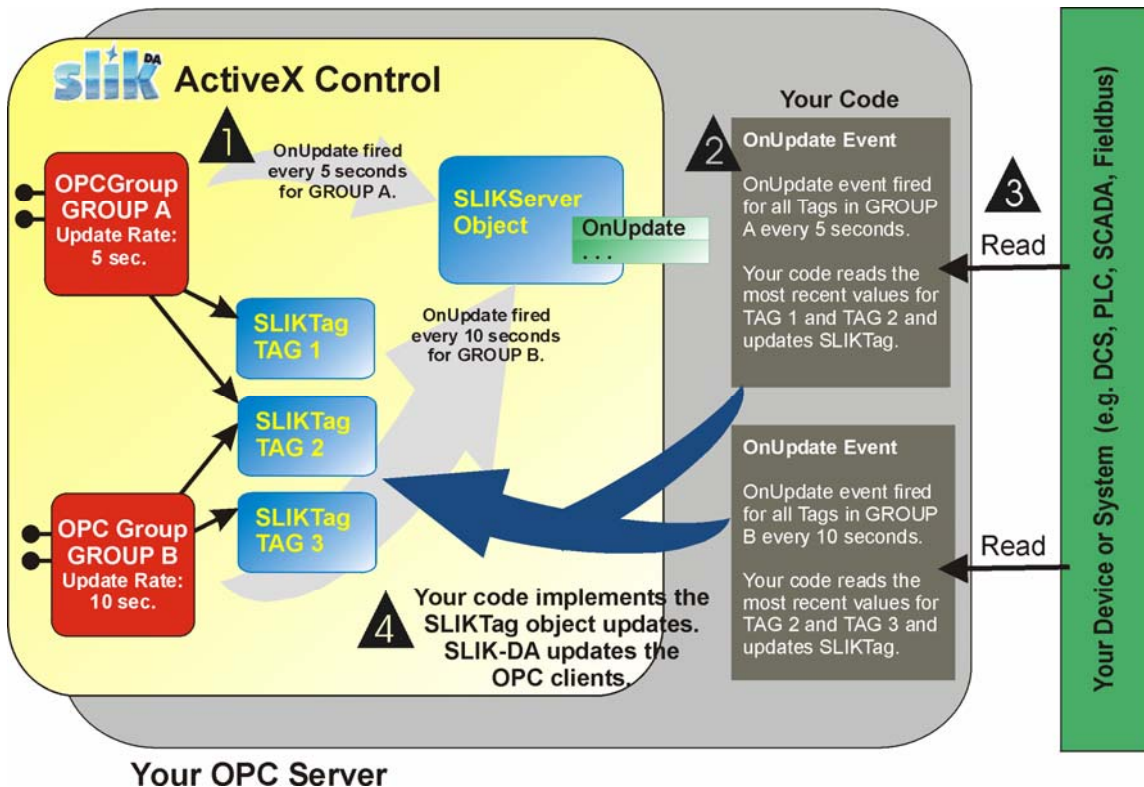
SLIK-DA Initiates the SLIKTag Object Updates

An OPC client organizes a server's resources using OPC Groups. After an OPC client connects to your server, it creates one or more OPC Groups. To each group the client adds one or more OPC Items. The client specifies an update rate parameter when creating an OPC Group. This parameter defines the fastest rate at which updates are to be sent from your server to the client.

By defining the optional *OnUpdate* event handler in your server, SLIK-DA initiates the update of your SLIKTag objects. Refer to the diagram and numbered steps shown below.

▲ The rate at which SLIK-DA initiates the update(s) is defined by the OPC groups currently active in your server. Each time a group's update period expires, the

(Continued on page 9)



Tutorial—Develop a Simple Server

(Continued from page 8)

OnUpdate event is fired for that group.

- ▲ The event handler is passed the list of Tags in the group.
- ▲ Your application code then reads the corresponding registers / points / database values.
- ▲ After completing the read request, your application code updates the corresponding SLIKTag objects with new values, qualities, and timestamps, as necessary. SLIK-DA takes care of updating the OPC clients.

TIP — Use the SLIKTag object's `UserData` property to store linking or cross-reference information between a SLIKTag and a device point.

The *OnUpdate* event update approach is typically used when communications with a system or device is very

fast.

Your Application Code Initiates the SLIKTag Object Updates

As your application acquires data from the underlying system or device, updates to values, qualities, and timestamps are propagated to the corresponding SLIKTag objects in the SLIKTags collection. SLIK-DA takes care of updating all the OPC clients if there are changes.

In general, your application will acquire data from your system or device in one of two ways, by polling or on an event basis.

Polling. Periodically, your application code reads data from your device or system via an API, protocol stack, or another ActiveX control (for example). The returned data is then processed and written to the corresponding SLIK-Tag objects.

Event Basis. Your system or device can provide notifications of data changes to your application. Your application processes the notifications and writes the data to the corresponding Tags in the SLIKTags collection.

Product Summary Sheet

System Requirements

Operating Systems — Windows 98, Windows NT 4.0 (SP5 or later), Windows 2000, Windows XP, Windows Server, Windows Vista.

OPC Clients Supported — All that are compliant with OPC Data Access 1.0a, 2.0, and 3.0 standards.

Development Environment — OLE automation enabled development tools like Visual Basic and Delphi.

Product Licensing

A SLIK-DA license is required for each development computer using the ActiveX control to create OPC servers. The OPC server you develop will only run if the server executable is generated on a SLIK-DA licensed computer. Re-distributables can be used in any number of products and no royalty fees are required. Annual developer support contracts are available.

Product Support

Northern Dynamic's dedicated support team is available via telephone and e-mail.

Your new license purchase includes 30 days of unlimited product support via email and telephone. The 30 days start from the date of first contact. The 30 day support period expires if not used within 180 days from the date of purchase.

Northern Dynamic's OPC experts are ready to assist with the development of your OPC server. Comprehensive on-site support and best-practices training are available. Contact a sales representative today for more information.

Proven Interoperability

SLIK-DA offers guaranteed compliance with the OPC Data Access standard. The reference implementation servers included with SLIK-DA are certified compliant with the OPC Foundation's Data Access Compliance Test Software.

Ordering Information

You can try SLIK-DA for free by downloading an evaluation version. When you are ready to purchase, contact Northern Dynamic to obtain your license. Your evaluation software can be converted to a licensed version immediately.

Contact Northern Dynamic for pricing information and a quotation.

telephone	+1 (519) 725-2071
	+1 (888) 265-7345 (toll free in NA)
fax	+1 (519) 725-2072
email	solutions@nordyn.com
website	http://www.OPCexperts.com

About Northern Dynamic

Our team of OPC experts can help you embed OPC technology into your software products, supply robust integration solutions, and provide third party support during your integration efforts.

Visit our website at www.OPCexperts.com to learn more about our OPC solutions.

- OPC Server Toolkits · OPC Gateway · OPC Consulting
- OPC Software Development · OPC Technical Support



Northern Dynamic Inc.
Suite 3—295 Hagey Blvd
Waterloo Research & Technology Park
Waterloo, Ontario
Canada N2L 6R5
Tel : +1 (519) 725-2071
Fax : +1 (519) 725-2072

