NERC PCS Project Update

NASPI Workgroup Meeting

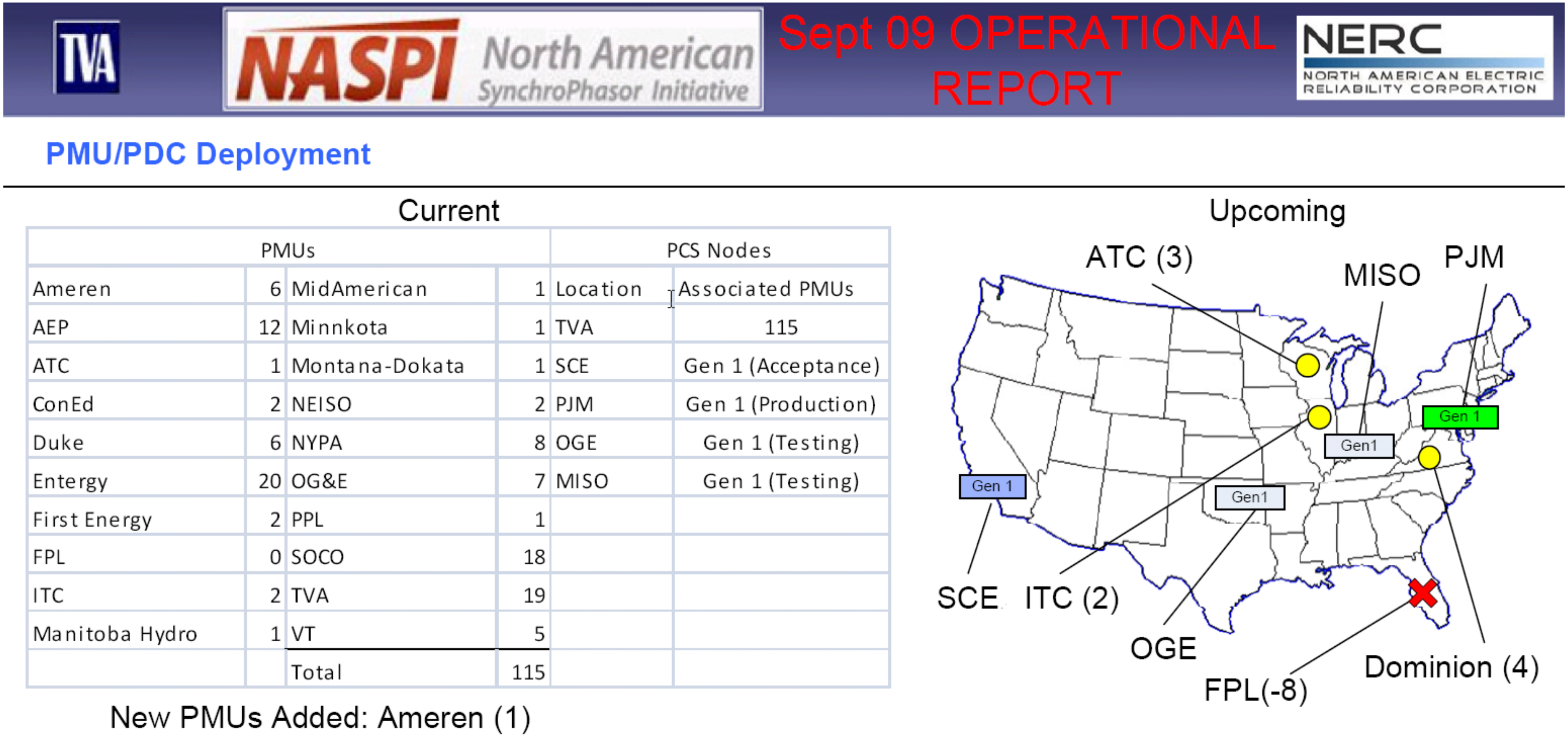
October 7th, 2009

Russell Robertson
Ritchie Carroll

Topics Covered

- **NERC PCS Project Status**
- **The openPDC Project**
- **Technical Project Update**

Example Operational Report



NASPI North American SynchroPhasor Initiative

Sept 09 OPERATIONAL REPORT

NERC North American Electric Reliability Corporation

PMU/PDC Deployment

Current

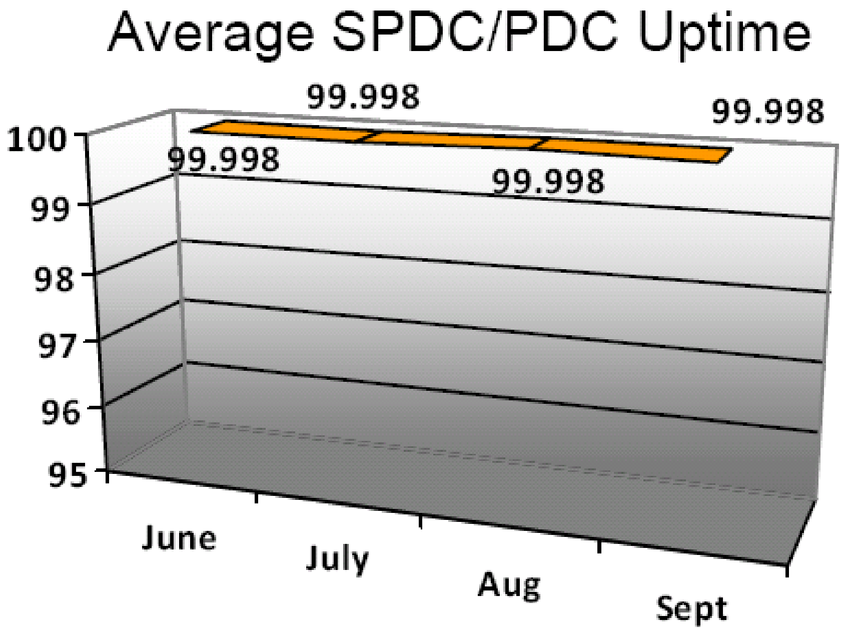
PMUs				PCS Nodes	
Ameren	6	MidAmerican	1	Location	Associated PMUs
AEP	12	Minnkota	1	TVA	115
ATC	1	Montana-Dokata	1	SCE	Gen 1 (Acceptance)
ConEd	2	NEISO	2	PJM	Gen 1 (Production)
Duke	6	NYPA	8	OGE	Gen 1 (Testing)
Entergy	20	OG&E	7	MISO	Gen 1 (Testing)
First Energy	2	PPL	1		
FPL	0	SOCO	18		
ITC	2	TVA	19		
Manitoba Hydro	1	VT	5		
		Total	115		

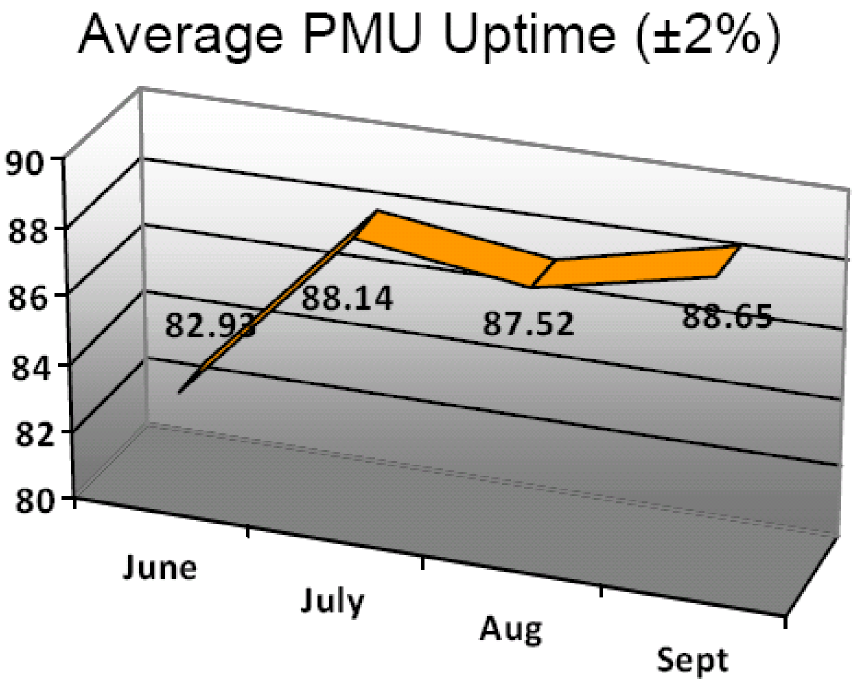
New PMUs Added: Ameren (1)

Upcoming

FPL is offline for NERC CIP upgrades.

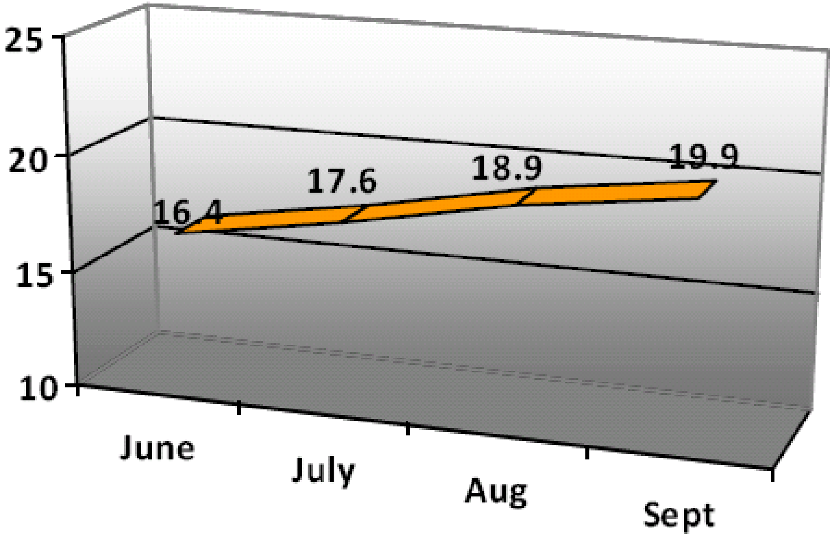
Uptime Statistics



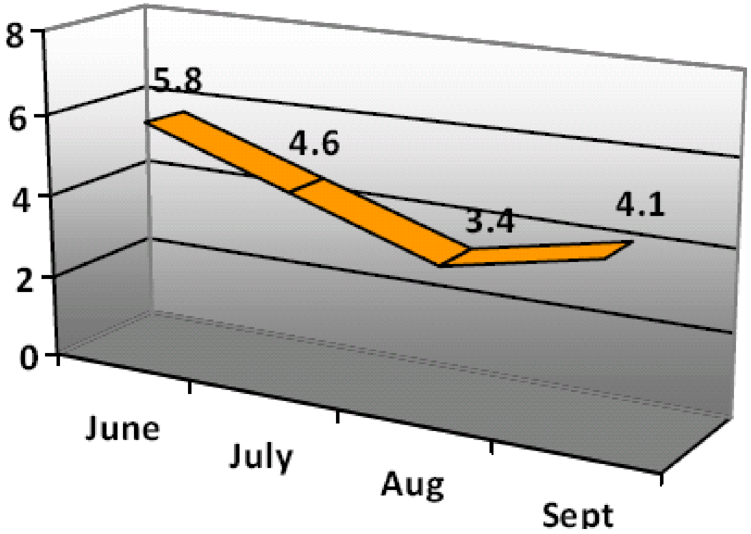


Storage Space

Archive Size (TB)



Free Space (TB)



NERC PCS Project - 5-Year Plan

- **2009**
  - Develop NERC PCS
  - Deploy 2 BETA NERC PCS Nodes
  - Set up Central Archive Infrastructure
- **2010**
  - Deploy 5 Production NERC PCS Nodes
  - Set up Communications Network
  - Operate / Grow Central Archive
- **2011**
  - Deploy 9 (4 new) Production NERC PCS Nodes
  - Integrate NASPInet into NERC PCS Network
  - Grow Communications Network
  - Operate / Grow Central Archive
- **2012**
  - Deploy new Production NERC PCS Nodes
  - Grow Communications Network
  - Operate / Grow Central Archive
- **2013**
  - Deploy new Production NERC PCS Nodes
  - Grow Communications Network
  - Operate / Grow Central Archive

2009 Deliverables and Status

- **Develop the NERC PCS**
  - System development is currently in progress.
  - TVA expects to start internal beta testing in late October.
- **Setup central archive (Hadoop)**
  - The test system has already been successfully deployed.
  - We expect to purchase production hardware starting in October.
- **Deploy pilot version of NERC PCS**
  - The contract for beta node hosts is being developed.
  - TVA is working with PJM to setup a VPN based LAN-to-LAN interim link.
  - The test system is expected to be deployed in early December.

Planned 2010 Deliverables

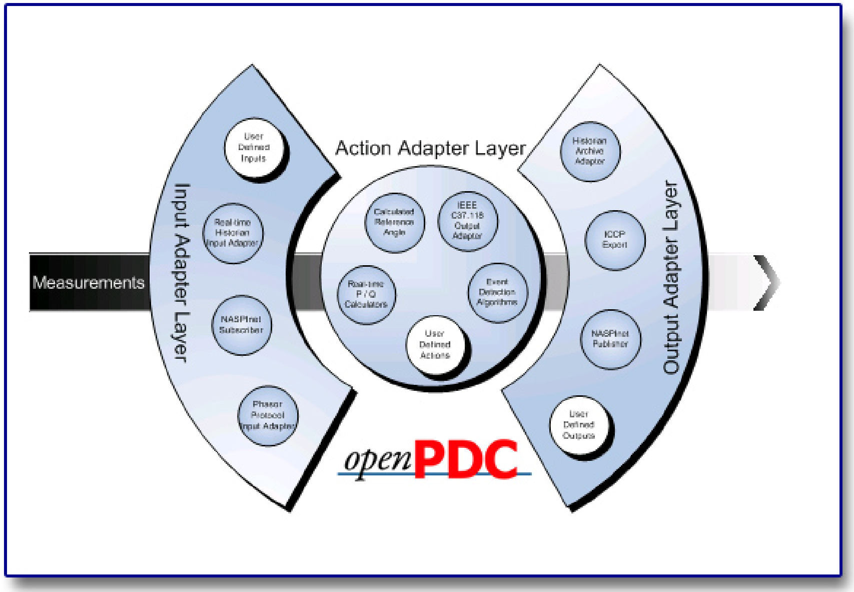
- **Complete PCS Security Assessment and implement recommendations**
- **Add 3rd beta NERC PCS Node** (April)
- **Formal release of NERC PCS system production version 1.0** (July)
- **NERC PCS comes online as all beta nodes are moved into production** (July)
- **Deployment of 4th NERC PCS node** (September)
- **Deployment of 5th NERC PCS node in** (November)
- **Add addition central archive storage** (as required)

Topics Covered

- NERC PCS Project Status
- The openPDC Project
- Technical Project Update

The OpenPDC Project

openPDC
Phasor Data Concentrator


Measurements
Input Adapter Layer
User Defined Inputs
Real-time Historian Input Adapter
NASPInet Subscriber
Phasor Protocol Input Adapter
Action Adapter Layer
Calculated Reference Angle
IEEE C37.118 Output Adapter
Real-time P / Q Calculators
Event Detection Algorithms
User Defined Actions
Output Adapter Layer
Historian Archive Adapter
ICCP Export
NASPInet Publisher
User Defined Outputs
openPDC

http://openPDC.codeplex.com

CodePlex
Open Source Project Community

- TVA has announced that the core of the NERC PCS (the TVA SuperPDC) will be open source
- Objective is to better support Stimulus/DOE plans to accelerate use of synchrophasors in the US
- Release will be made under a liberal open source agreements the supports vendor commercialization
- Key design element is the ability for others to easily extend the openPDC at the input, action or output layer
- TVA's hope is that this code will be an enabling technology for the industry and vendors at the international level
- TVA will be seeking partnerships to help extend and improve the openPDC

openPDC Phasor Data Concentrator What's Included...

- **The TVA Code Library:** this consists of code for handling sockets, threading, error management, and security plus basic data concentration and archival and general phasor protocol parsing and recomposition.
- **The openPDC synchrophasor projects:** these include all the TVA SuperPDC components plus the TVA PMU Connection Tester.
- **An experimental *prototype* implementation of NASPInet** is in the code in its nascent stages to spur further development and discussion.

openPDC Phasor Data Concentrator Source Code Status

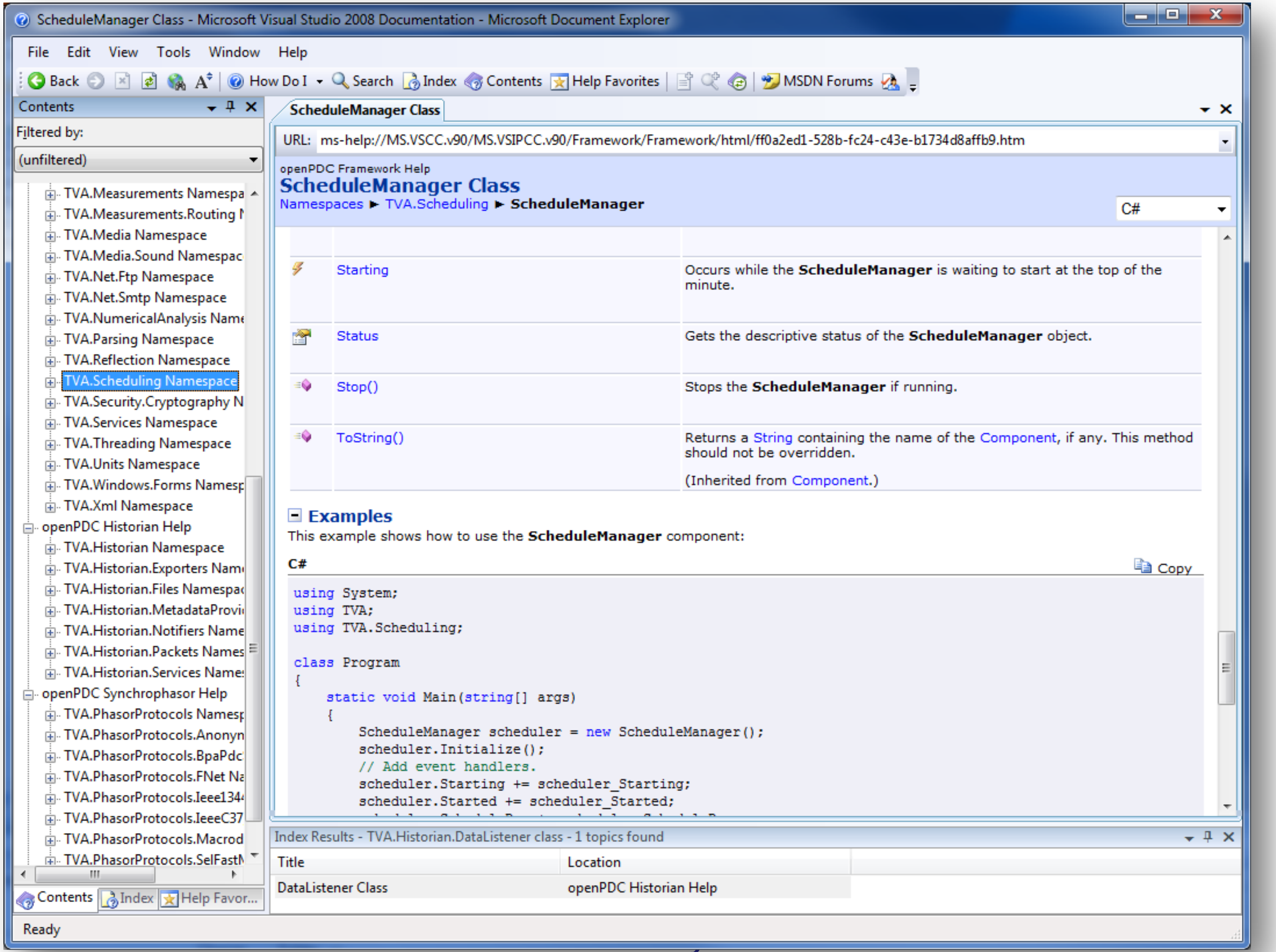
- Code based on TVA's SuperPDC that has been in production use since 2004.
- The source code is based on .NET 3.5 and written almost exclusively in managed C#.
- The code set has over 300,000 lines of fully documented code spanning 630+ classes.
- Includes detailed API help files for the source code.
- Everything needed to create your own operational PDC is included in the code.

And… **the openPDC manager, a web based remote configuration tool, is under development and scheduled for beta release November 1**.

Priorities on Documentation

- **Lines of code vs. documentation**
  - Of the 300,000 lines of code, 6% are structured API comments which are used to create the integrated help.
  - To give you an idea of the amount of help this is - the API help will fill out over 4600 printed pages.
- **The four priority levels**
  - API Level Documentation - *completed*
  - Getting Started Documentation - *completed*
  - System Documentation – in progress
  - Operational Documentation – not started

Help can be integrated within Visual Studio


ScheduleManager Class - Microsoft Visual Studio 2008 Documentation - Microsoft Document Explorer
File
Edit
View
Tools
Window
Help
Back
A⁺
How Do I
Search
Index
Contents
Help Favorites
MSDN Forums
Contents
Filtered by:
(unfiltered)
TVA.Measurements Namespa
TVA.Measurements.Routing N
TVA.Media Namespace
TVA.Media.Sound Namespac
TVA.Net.Ftp Namespace
TVA.Net.Smtp Namespace
TVA.NumericalAnalysis Name
TVA.Parsing Namespace
TVA.Reflection Namespace
TVA.Scheduling Namespace
TVA.Security.Cryptography N
TVA.Services Namespace
TVA.Threading Namespace
TVA.Units Namespace
TVA.Windows.Forms Namesp
TVA.Xml Namespace
openPDC Historian Help
TVA.Historian Namespace
TVA.Historian.Exporters Nam
TVA.Historian.Files Namespac
TVA.Historian.MetadataProvi
TVA.Historian.Notifiers Name
TVA.Historian.Packets Names
TVA.Historian.Services Name
openPDC Synchrophasor Help
TVA.PhasorProtocols Namesp
TVA.PhasorProtocols.Anonyn
TVA.PhasorProtocols.BpaPdc
TVA.PhasorProtocols.FNet Na
TVA.PhasorProtocols.Ieee134
TVA.PhasorProtocols.IeeeC37
TVA.PhasorProtocols.Macrod
TVA.PhasorProtocols.SelFastN
Contents
Index
Help Favor...
ScheduleManager Class
C#
Copy

URL: ms-help://MS.VSCC.v90/MS.VSIPCC.v90/Framework/Framework/html/ff0a2ed1-528b-fc24-c43e-b1734d8affb9.htm

openPDC Framework Help

ScheduleManager Class

Namespaces ▶ TVA.Scheduling ▶ **ScheduleManager**

Starting	Occurs while the **ScheduleManager** is waiting to start at the top of the minute.
Status	Gets the descriptive status of the **ScheduleManager** object.
Stop()	Stops the **ScheduleManager** if running.
ToString()	Returns a String containing the name of the Component, if any. This method should not be overridden. (Inherited from Component.)

Examples

This example shows how to use the **ScheduleManager** component:

C#

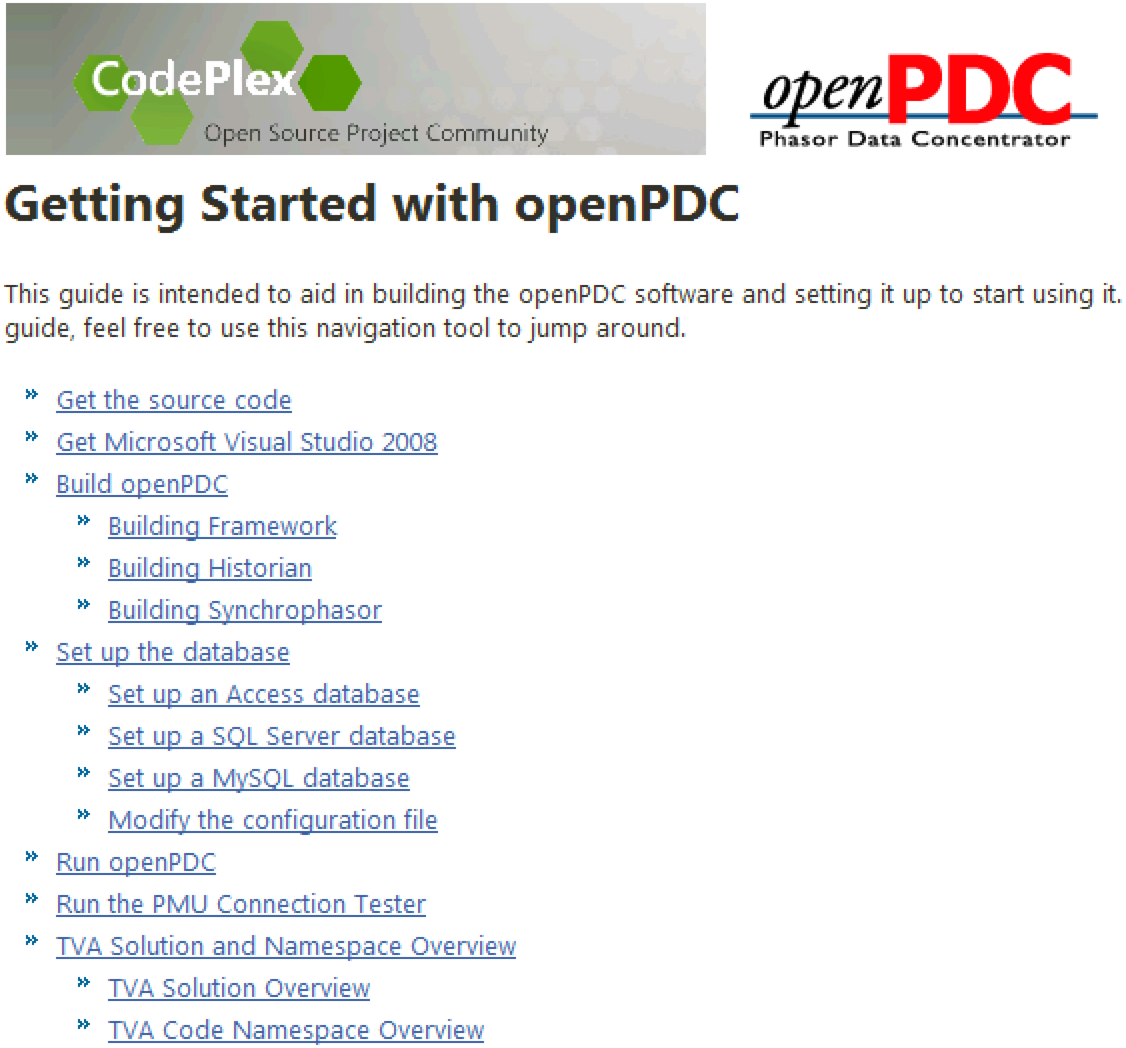
```
using System;
using TVA;
using TVA.Scheduling;

class Program
{
    static void Main(string[] args)
    {
        ScheduleManager scheduler = new ScheduleManager();
        scheduler.Initialize();
        // Add event handlers.
        scheduler.Starting += scheduler_Starting;
        scheduler.Started += scheduler_Started;
```

Index Results - TVA.Historian.DataListener class - 1 topics found

Title	Location
DataListener Class	openPDC Historian Help

Ready

Step-by-step instructions are provided



Getting Started with openPDC

This guide is intended to aid in building the openPDC software and setting it up to start using it.
guide, feel free to use this navigation tool to jump around.

- Get the source code
- Get Microsoft Visual Studio 2008
- Build openPDC
  - Building Framework
  - Building Historian
  - Building Synchrophasor
- Set up the database
  - Set up an Access database
  - Set up a SQL Server database
  - Set up a MySQL database
  - Modify the configuration file
- Run openPDC
- Run the PMU Connection Tester
- TVA Solution and Namespace Overview
  - TVA Solution Overview
  - TVA Code Namespace Overview

openPDC Phasor Data Concentrator The Pre-Release Review

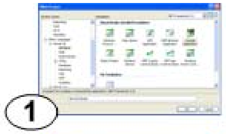
- **Companies Participating:**
  - Information Trust Institute (University of Illinois)
  - Pacific Northwest National Labs
  - AREVA T&D
- **Reviewers were selected from a university, a national lab and a vendor to get a balanced set of views.**
- **Review was conducted by these independent entities to help find any issues and provide insight into possible improvements prior to release.**

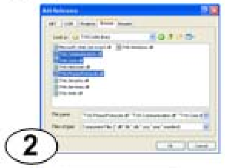
ITI/Illinois Preliminary Review

- **Preliminary Review Completed Oct 2, 2009**
  - ~50 man-hours
  - Mostly manual with some help from tools
- **Preliminary review focused on compatibility and security**
  - Several issues identified in input/bounds checking, access control, information leakage, crypto
  - Preliminary review report submitted to TVA
- **Identified important topics for further review**

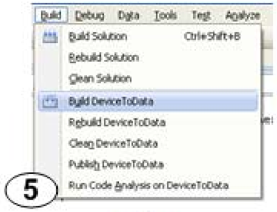
TVA is actively addressing these topics; some of which have already been resolved.

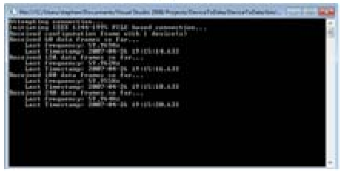
Device to Data in Five Easy Steps





```
// Define the connection string
parser.ConnectionString =
    [illegible]
    [illegible]
```





1. Create a project
2. Add references
3. Copy in the code snippet
4. Set up your data source
5. Run the application

(6) Take the afternoon off.

Where do I get the code?

CodePlex
Open Source Project Community

openPDC
Phasor Data Concentrator

http://openpdc.codeplex.com

- Any public contributions consisting of new features, updates or fixes will be accepted by TVA.
- All contributions will be reviewed for security and completeness before they are integrated with the public code.
- Copyrights and attributions to contributors will remain in the code.
- Register (free) to be notified via email of code updates and improvements.

openPDC Phasor Data Concentrator What's Happening Now...

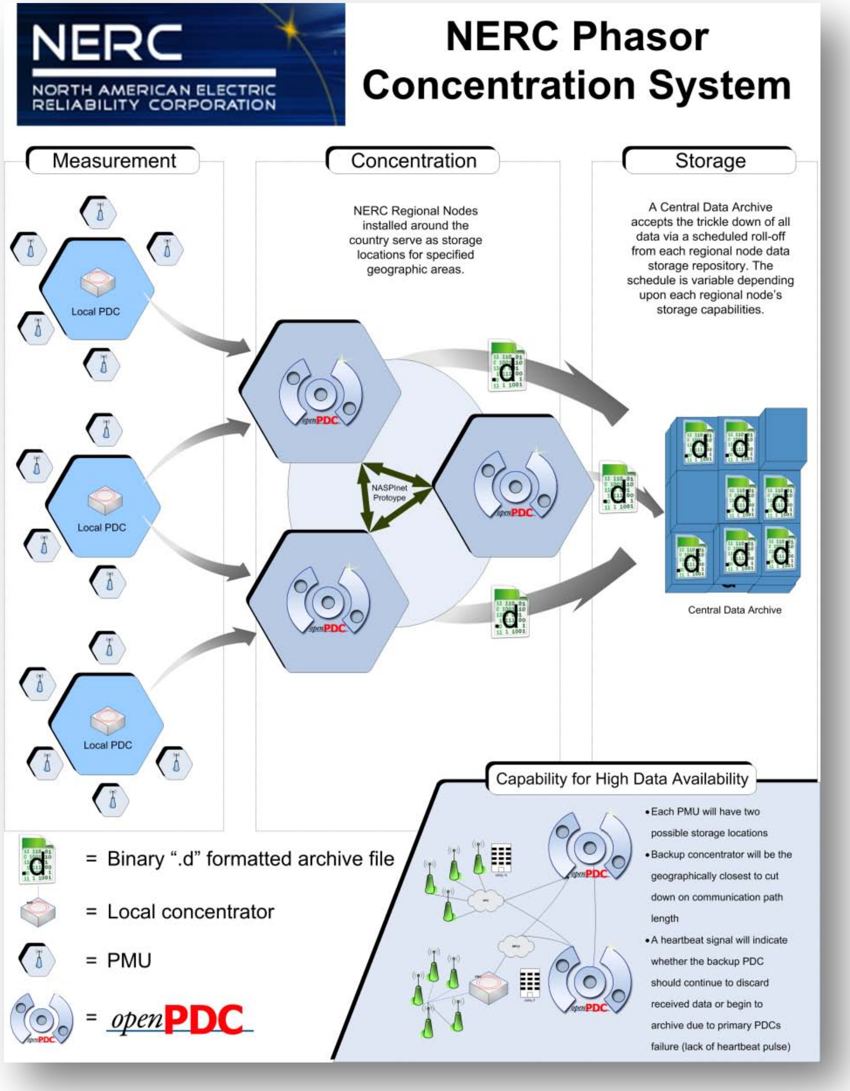
- **We are busy developing the NERC PCS;** however, the NERC PCS is based on this open source code. Any changes, updates or bug fixes we make to the code will be made available to you.
- **We'll be encouraging other developers to add to the open source code base.** To accommodate more developers, we are adding unit tests to the code so that changes made by others can be tested for consistency.
- **An initial version of the "Getting Started Guide" and FAQ have been posted to the website.** As we get more questions, we will answer these and post them to the FAQ.

Topics Covered

- **NERC PCS Project Status**
- **The openPDC Project**
- **Technical Project Update**

NERC PCS Technical Progress

- The NERC PCS has the openPDC as its core technology
- The core has been wrapped with NERC proprietary elements
- An experimental “NASPInet-like” protocol for inter-node communication is in development.
- Code is well along in development for transfer of regional node data to the central archive



NERC Regional Nodes installed around the country serve as storage locations for specified geographic areas.

A Central Data Archive accepts the trickle down of all data via a scheduled roll-off from each regional node data storage repository. The schedule is variable depending upon each regional node’s storage capabilities.

= Binary “.d” formatted archive file

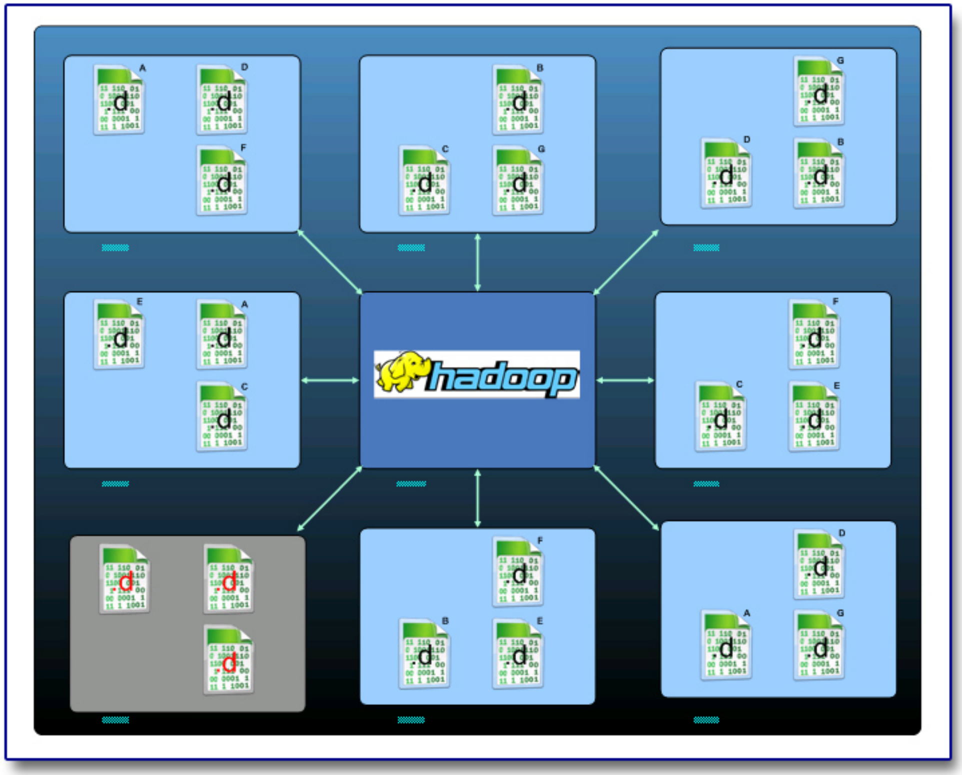
= Local concentrator

= PMU

= openPDC

- Each PMU will have two possible storage locations
- Backup concentrator will be the geographically closest to cut down on communication path length
- A heartbeat signal will indicate whether the backup PDC should continue to discard received data or begin to archive due to primary PDCs failure (lack of heartbeat pulse)

The NERC PCS Central Archive

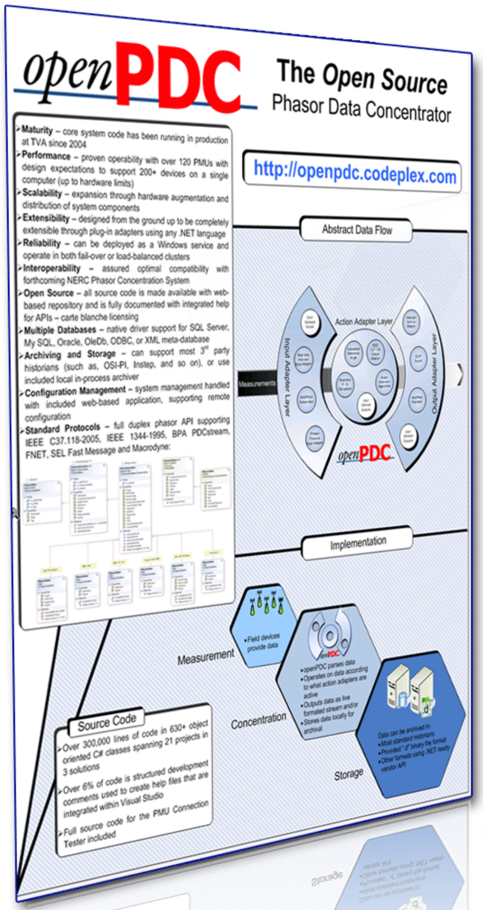


- A test system has been deployed and is operational.
- This test system is based on “Hadoop” technology.
- Hadoop is an open source product that uses low cost commodity hardware to redundantly store data.
- Hadoop also enables parallel processing of massive amounts of data.
- We are currently developing the recoverable data transport process for moving data from a “Regional Node” into the central archive.

Hadoop Pilot Operational

- **Pilot hardware installed and operational**
- **Total cost ~$20k** (direct cost $16.5k)
- **Storage:**
  - 48 TB Physical
  - 14 TB Effective (3 way replication)
- **Capable of 109 MB/sec write speeds per drive**

Poster / Handouts – To explain more



openPDC The Open Source Phasor Data Concentrator

- **Maturity** – core system code has been running in production at TVA since 2004
- **Performance** – proven operability with over 120 PMUs with design expectations to support 200+ devices on a single computer (up to hardware limits)
- **Scalability** – expansion through hardware augmentation and distribution of system components
- **Extensibility** – designed from the ground up to be completely extensible through plug-in adapters using any .NET language
- **Reliability** – can be deployed as a Windows service and operate in both fail-over or load-balanced clusters
- **Interoperability** – assured optimal compatibility with forthcoming NERC Phasor Concentration System
- **Open Source** – all source code is made available with web-based repository and is fully documented with integrated help for APIs – carte blanche licensing
- **Multiple Databases** – native driver support for SQL Server, My SQL, Oracle, OleDb, ODBC, or XML meta-database
- **Archiving and Storage** – can support most 3rd party historians (such as, OSI-PI, Instep, and so on), or use included local in-process archiver
- **Configuration Management** – system management handled with included web-based application, supporting remote configuration
- **Standard Protocols** – full duplex phasor API supporting IEEE C37.118-2005, IEEE 1344-1995, BPA PDCstream, FNET, SEL Fast Message and Macrodyne:

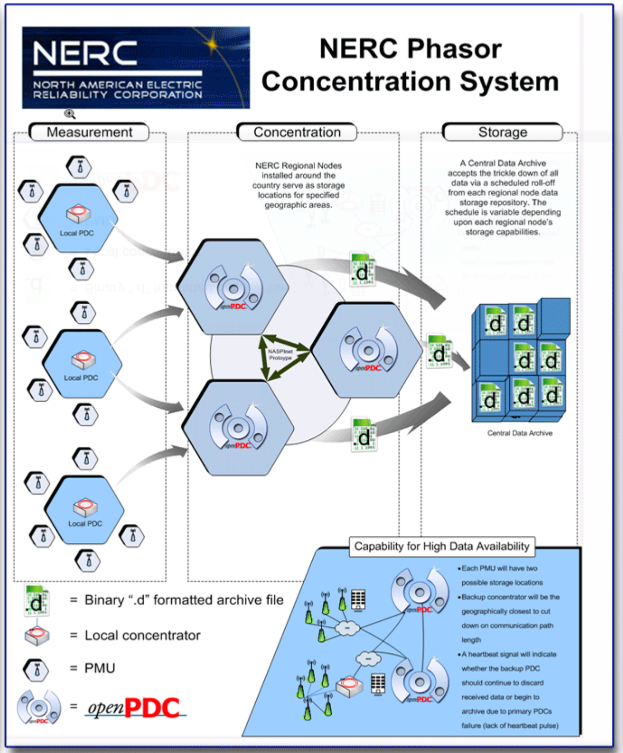
http://openpdc.codeplex.com

- Field devices provide data
- openPDC parses data
- Operates on data according to what action adapters are active
- Outputs data as live formated stream and/or
- Stores data locally for archival

Data can be archived in
- Most standard historians
- Provided ".d" binary file format
- Other formats using .NET ready vendor API

Source Code

- Over 300,000 lines of code in 630+ object oriented C# classes spanning 21 projects in 3 solutions
- Over 6% of code is structured development comments used to create help files that are integrated within Visual Studio
- Full source code for the PMU Connection Tester included



NERC NORTH AMERICAN ELECTRIC RELIABILITY CORPORATION

NERC Phasor Concentration System

Measurement
Concentration
Storage
Local PDC
Local PDC
Local PDC
NASPInet Protoype
Central Data Archive

NERC Regional Nodes installed around the country serve as storage locations for specified geographic areas.

A Central Data Archive accepts the trickle down of all data via a scheduled roll-off from each regional node data storage repository. The schedule is variable depending upon each regional node's storage capabilities.

.d = Binary ".d" formatted archive file

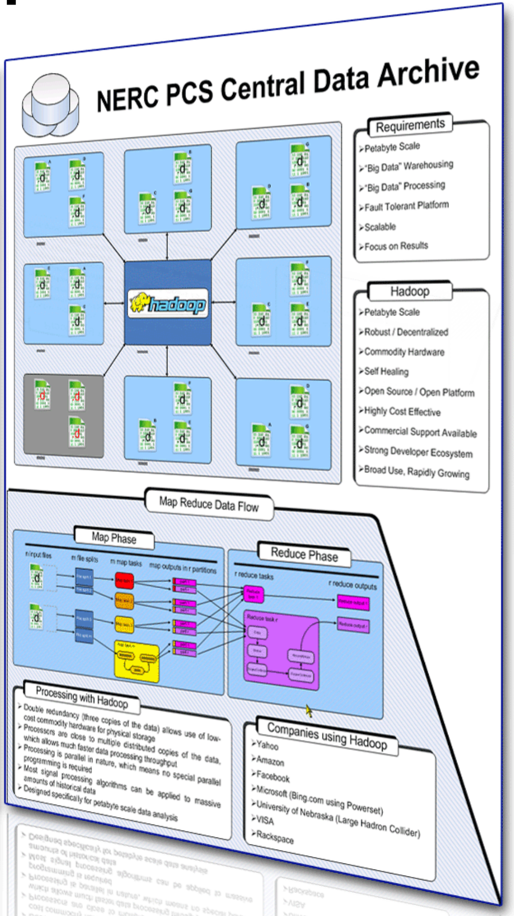
= Local concentrator

= PMU

= openPDC

Capability for High Data Availability

- Each PMU will have two possible storage locations
- Backup concentrator will be the geographically closest to cut down on communication path length
- A heartbeat signal will indicate whether the backup PDC should continue to discard received data or begin to archive due to primary PDCs failure (lack of heartbeat pulse)



NERC PCS Central Data Archive

Requirements

- Petabyte Scale
- "Big Data" Warehousing
- "Big Data" Processing
- Fault Tolerant Platform
- Scalable
- Focus on Results

Hadoop

- Petabyte Scale
- Robust / Decentralized
- Commodity Hardware
- Self Healing
- Open Source / Open Platform
- Highly Cost Effective
- Commercial Support Available
- Strong Developer Ecosystem
- Broad Use, Rapidly Growing

Processing with Hadoop

- Double redundancy (three copies of the data) allows use of low-cost commodity hardware for physical storage
- Processors are close to multiple distributed copies of the data, which allows much faster data processing throughput
- Processing is parallel in nature, which means no special parallel programming is required
- Most signal processing algorithms can be applied to massive amounts of historical data
- Designed specifically for petabyte scale data analysis

Companies using Hadoop

- Yahoo
- Amazon
- Facebook
- Microsoft (Bing.com using Powerset)
- University of Nebraska (Large Hadron Collider)
- VISA
- Rackspace