EPICS TO TINE TRANSLATOR RELEASE 2.0

Philip Duval, Matthias Clausen, Zoltan Kakucs
Deutsches Elektronen Synchrotron, Germany

Abstract

The initial version of the TINE translator for EPICS [1] has so far proved a great success as DESY. The initial needs of providing data from an EPICS [2] IOC via TINE [3] were such that the read-only nature of the translator were all that were needed. Likewise the initial name mapping scenario where the 28-character Channel Access name was simply mapped one-to-one to the TINE device name and the 4-character Channel Access field name was mapped to the TINE property name was deemed adequate, especially as more acceptable names were frequently provided via the translator's composite system.

In this paper, we report on the new features available in the EPICS to TINE translator. In particular, the ability of the translator to accept WRITE access commands (EPICS dbput()s) and the mapping of alarms will be presented. More importantly, the newer more logical name mapping will be discussed, whereby the Channel Access name is parsed into a more meaningful combination of device name and device property.

As the initial translator was used primarily on VxWorks IOCS, we shall also give results for other platforms.

INTRODUCTION

By and large accessing EPICS IOCs (Input/Output Controllers) via TINE has proven a great success. The use of EPICS in HERA accelerator control is itself marginal, where the vast majority of controllable devices are based on TINE. Nonetheless some subsystems such as the superconducting electron RF cavities are controlled by EPICS, as are the power and cooling subsystems. The initial version of the EPICS to TINE translator was designed to address the immediate needs of acquiring and integrating HERA data otherwise only available via Channel Access. That is, the translator offered a 'read-only' view of the EPICS IOC. Any changing of settings was made via channel access from an EPICS control pannel. It was not long before the need to send commands to an EPICS IOC via TINE was evident. Furthermore, issues such as mapping EPICS alarms into TINE alarms were also cropping up. It soon become clear that a fully functional EPICS to TINE translator was required. At the same time we decided that a more viable name mapping from the flat channel access name to the TINE name hierarchy could be realized. To be sure, the use of 'Composites' or TINE alias lists can be used to try to rationalize the channel access namespace. However, the name mapping presented in the first release suffers from using channel access fields as TINE properties and thereby confusing 'meta' properties with properties. The scheme presented here will be seen to be more logical in its approach.

IMPROVEMENTS

Commands

Opening up the EPICS to TINE translator to commands was a fairly straightforward, if not trivial, exercise. Essentially, it involves demanding the TINE WRITE access bit is set (which is in fact the definition of a TINE command), reformatting the incoming TINE data as a string and calling dbPutField() with EPICS format DBR_STRING (or DBR_ENUM if necessary). TINE calls which atomically WRITE and then READ follow the call to dbPutField() with dbGetField(). In reality this is only apparently atomic, as the time interval between the two calls is indeterminate. It is perhaps instructive to point out one of the differences in data exchange philosophy between EPICS and TINE. Where EPICS database-centric, dealing in 'puts' and 'gets', TINE is data link-centric, where a data link can have an input and an output data set and the nature of the transaction is governed by the TINE property, which in fact might not involve any data exchange at all (for example, properties such as 'RESET' or 'INITIALIZE').

Note that by using the EPICS calls dbPutField() and dbGetField() we are bypassing channel access security, but not TINE security. This is a design decision. When the EPICS to TINE translator is in place it seems natural to allow TINE commands to adhere to TINE security and channel access commands to adhere to channel access security. On the one hand, this might be introducing a potential security leak, where one set of security attributes could be more lax than the other. On the other, applying both sets is more likely introducing an administrative headache and source of confusion. One could in principle apply both sets of security, perhaps optionally via a configuration parameter..

Alarms

The EPICS threshold alarms are likewise fairly straightforward to reproduce. It was decided that the best way to 'see' these alarms via TINE was to regenerate them. Namely, at initialization time, the EPICS database is queried for its alarm content concerning the 'HI', 'HIHI', 'LO', and 'LOLO' fields. This information is then passed on to the TINE local alarm server by appending the alarm watch table. What this means is that the TINE translator will itself be queried by the local alarm server to determine if a value is near a threshold. That is the EPICS alarms are themselves not mapped, but regenerated by the TINE subsystem.

The remaining issue when dealing with alarm translation is the mapping of the alarm severities. A TINE alarm can have one of 15 different severities ranging from 0 (ignored) to 15 (fatal) with various degrees of information and warnings in between. An EPICS alarm can have one of 4 severities, the most severe being 4. So it seemed logical to use a 'times four minus one' style mapping.

Name Mapping

The mapping of channel access names into the TINE name space has always been a challenge. The EPICS namespace is flat, and a standard channel access name consists of 28 characters plus a 4 character field. The EPICS database programmer will think of something meaningful to apply to the initial 28 characters, whereas the EPICS field will be a systematic identifier. So unless there is some site-wide naming strategy which is strictly adhered to, the channel name is often left to the whim of the database programmer. The TINE namespace is hierarchical, where a device context (16 characters) and a device server name (16 characters) uniquely identify a TINE FEC (Front End Controller) and a device name (16 characters) specifies a particular device on the FEC, i.e. 'what it is'. A device property (32 characters) identifies the request which is made to the device, i.e. 'what it does'

In the initial release of the EPICS to TINE translator,. as there was no appearance of any systematics within the channel access names in use at DESY, it was decided to treat the initial portion of the channel name as the TINE device name and use the EPICS fields as the TINE properties. This created a slight problem in that a normal TINE device name contains only 16 characters. Using only the 4 characters of the EPICS field for the TINE property is in this case a decided waste of buffer space. Fortunately, TINE allows 'long device names', effectively reversing the size limitations of the TINE device name and the TINE property. While this works, it presents by itself an 'unusual' view of a TINE device server. In the past, the 'composites' feature of the EPICS to TINE translator has been used not only to make collections of EPICS channels, but to provide user friendly names to the client-side developers. The device context and device server name of the EPICS IOC are derived entirely from the TINE fecid.csv startup file.

In the second release of the translator we have tried to make use of the tendency within a channel access name to use a colon ':' as a separator, as to describe 'what it is' in the first part of the channel name and 'what it does' in the second part. Therefore at initialization time, we can parse the channel names from the end to the beginning looking for the first separator, thereby splitting the channel name into two pieces, the second of which will be registered as a property and the first part as a device belonging to this property. In TINE parlance this means that the resulting server will have 'property-query' precedence, and will inform a TINE browser that the device list should be queried following a change in property. It also means that the EPICS fields no longer show up as properties, but are instead treated as TINE 'meta' properties.

By taking this approach, the reliance on 'long device names' is also no long necessary, as the name 'splitter' will ensure that the device name is indeed 16 characters or less. If the channel name consists of 28 characters without a separator or if the left-hand part of the split contains more than 16 characters without a separator, the entire name will show up as a TINE property.

For instance, a channel access name such as "AL:K:KV:A1:UL1L2TA1_ai.VAL" will be mapped into the TINE property "UL1L2TA1_ai" applied to the TINE device "AL:K:KV:A1". The field ".VAL" is superfluous in this case and in the absence of any other meta property qualifier will be automatically appended to the channel name constructed from a TINE call, before accessing dbPutField() or dbGetField(). A TINE alias file could also offer more human readable names for someone browsing the control system.

STATUS AND CONCLUSIONS

The EPICS to TINE translator has been used without problems on many EPICS IOCs at DESY for the past few years. It is standard equipment on EPICS IOCs relating to HERA, specifically the superconducting electron RF cavities and most recently the FEEDBOX data readout for the ZEUS experiment. It is likewise used on EPICS IOCS relating to the power and cooling infrastructure not only for HERA but for the pre-accelerator chain. In particular, it is the desire for keeping better operation statistics in the pre-accelerator chain that motivated the EPICS to TINE alarm translation in Release 2.0. The EPICS to TINE translator is also being used to some extent at DESY Zeuthen, where in fact the need to allow a TINE command to map to dbPutField() first arose.

Although the majority of EPICS IOCS at DESY are VxWorks IOCS we should point out that the TINE translator runs fine on the other platforms where EPICS runs.

REFERENCES

[1] "An EPICS to TINE Translator", Z. Kakucs, P. Duval, M. Clausen, ICALEPCS 2001.
[2] http://www.aps.anl.gov/epics
[3] http://desyntwww.desy.de/tine