

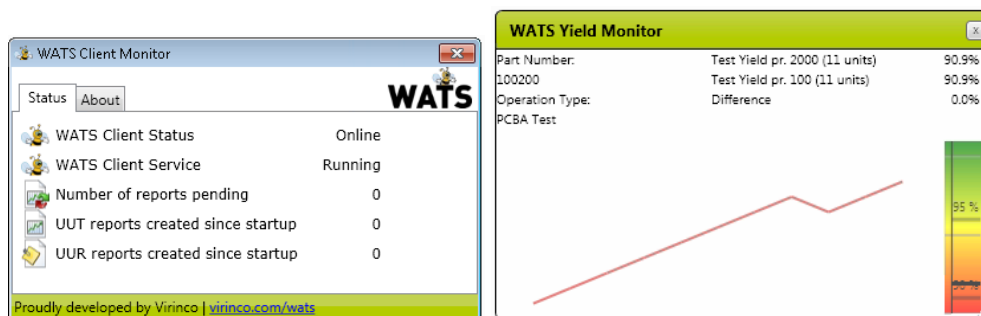


Client



WATS Client

The WATS client is the software installed on a computer (ATE) and communicates with a WATS server. The main purpose of the client is to communicate data from applications locally and safely transfer it to the WATS system - from any location - using internet communication (TDM reporting supports offline connection).



WATS Client Monitor with YIELD chart to instant monitor running production.

The WATS Client is the first line of storage and SPC in the WATS system and utilizes the WATS Transfer Technology to propagate data to the next level. The application can either install as an add-on to NI TestStand, or as an extension to your test application utilizing the LabVIEW toolkit or .NET API.

The WATS Client also supports indirect interfacing using a Translator plug-in structure (reading flat files and converting to WATS reports).



NI TestStand add-on

WATS adds new functionality and integrates seamlessly with TestStand. A fresh WATS Client installation, install - configure - log UUT reports to the WATS system, takes **less than 5 minutes** (assuming TestStand default settings).

Learn more about NI TestStand here: www.ni.com/teststand



NI LabVIEW toolkit

Use the WATS LabVIEW toolkit and take advantage of the powerful WATS reporting functionality. Implement the “easy-to-use” toolkit into your test application, without concerns regarding database connections, SQL statements or flat files handling.



.NET API

Take full control of the WATS client using the embedded .NET API! Any programming language that can call a .NET assembly can utilize the API, allowing tight integration with the different WATS Modules.



Translator plug-in

Using C#, customers can create their own Translator plug-in to interface flat files on the machine. Using this method, there is a clean cut between your application and WATS. Virinco have a set of plug-in templates that can be used. This method limits interaction with WATS Modules and is best suited for strict test and repair reporting.

Supported Modules and Functionality

