

Microsoft® CCR and DSS Toolkit 2008

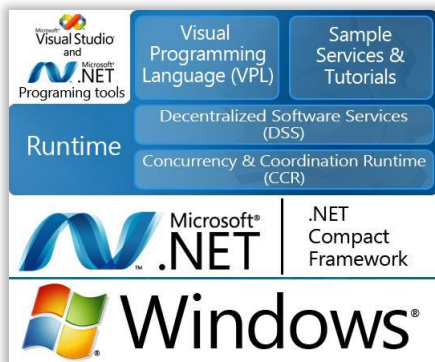
Concurrency & Coordination Runtime and Decentralized Software Services

The Microsoft® CCR and DSS Toolkit 2008 delivers a set of .NET and Compact Framework class libraries and tools that enable developers to better deal with the inherent complexities in creating loosely-coupled concurrent and distributed applications.

The Toolkit is designed to help developers take advantage of the Concurrency and Coordination Runtime (CCR) and Decentralized Software Services (DSS) originally released as part of Microsoft Robotics Developer Studio.

Microsoft CCR and DSS Toolkit 2008 targets early adopters by providing access to select technologies today; transitioning to Microsoft's .NET Framework in the future.

To learn more about Microsoft CCR and DSS Toolkit 2008, visit:
<http://www.microsoft.com/ccrdss>



Components included in Microsoft RDS are shaded in blue. Other components must be acquired separately. Additional charges may apply.

Problem Areas Addressed

Microsoft CCR and DSS Toolkit can help developers by targeting two significant challenges common in today's application development:

- Supporting multi-core and concurrent applications by facilitating asynchronous operations, dealing with concurrency, exploiting parallel hardware and handling partial failure.
- Supporting robust, distributed applications based on a lightweight state-driven service composition, event notification, and loose coupling through data-isolation.

Customer Scenarios

Existing scenarios deploying Microsoft CCR and DSS Toolkit 2008 have some or all of the following characteristics:

- The application is data-driven with a need for processing data asynchronously.
- The application consists of data-flow oriented services composed in a loosely coupled manner either within a single node or across the network.
- A single application can be distributed on heterogeneous devices ranging from servers to embedded devices.

Coordination & Concurrency

Microsoft CCR and DSS Toolkit 2008 provides developers an easy way to create and coordinate loosely-coupled, data-driven applications:

- Simple to integrate into asynchronous data-driven applications

```
static IEnumerable<ITask> CopyStream(Stream source, Stream dest)
{
    var buffer = new byte[4096];
    int read = 0;
    do
    {
        PortSet<int,Exception> readResult = StreamAdapter.Read(
            source, buffer, 0, buffer.Length);
        yield return readResult.Choice();
        var exception = (Exception)readResult;
        if (exception != null)
            yield break;
        read = (int)readResult;
        var writeResult = StreamAdapter.Write(buffer,0,read);
        yield return writeResult.Choice();
    } while (...)
```

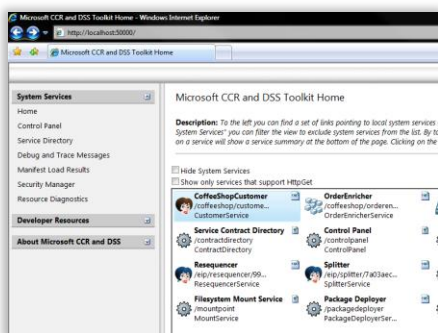
Concise asynchronous code

- Maximizes the utilization of processing resources by automatically load-balancing across multiple cores
- Eliminates the complexities of manual threading, locks, and semaphores
- Provides powerful in-memory message-passing architecture with powerful orchestration primitives enabling coordination of data and work
- Event based notification model provides increased efficiency in processor and network utilization
- Increased resiliency through flexible failure handling and late-binding

State-Driven and Composable

CCR and DSS Toolkit 2008 enables the creation of state-driven and composable loosely-coupled applications:

- Lightweight state-oriented services architecture enabling late-bound, loosely coupled applications
- Built-in support for composing, observing, and managing services



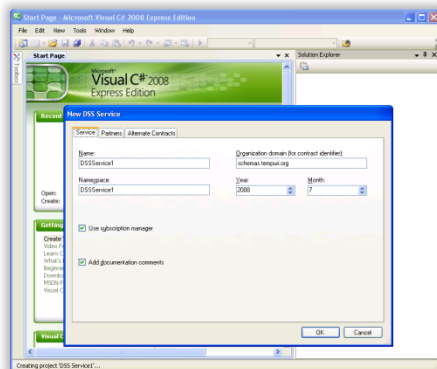
Observing and interacting with the web browser¹

- Integrated event-notification model that communicates state changes between services
- Consistent model between local and distributed applications
- Separation of code and state enables flexible user interface
- Windows®, Web, & Web-services compatible

Flexible Development

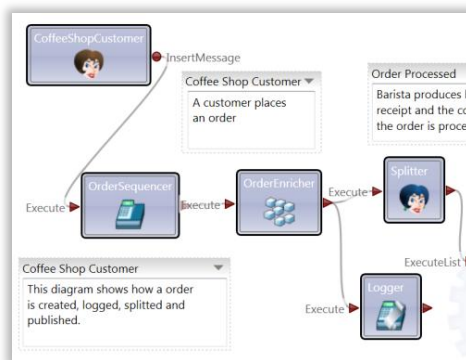
CCR and DSS Toolkit 2008 supports a wide range of options for developing and deploying loosely-coupled concurrent and distributed applications:

- Support for Visual Studio® product family and .NET programming tools



Integrated Development with Visual Studio

- Simple drag-and-drop visually declarative programming language for easy application composition



VPL makes it easy to create applications¹

- Simple drag-and-drop application configuration and deployment tool
- Support for .NET Framework and .NET Compact Framework libraries
- Sample services, tutorials and documentation make it easy to get started

System Requirements

Application development and runtime deployment is supported on the following operating systems:

- Desktop OS: Windows® XP (all editions including x64), Windows Vista® (all editions including x64).
- Server OS: Windows Server® 2003 R2 (all editions including x64); Windows Server® 2008 (all editions including x64).

Runtime deployment is only supported on the following operating systems:

- Windows® XP Embedded, Windows® Embedded CE 5.0, Windows® Embedded CE 6.0

The Toolkit requires and installs:

- .NET Framework 3.5 SP1
- .NET Compact Framework 3.5

¹ Screenshots were modeled with permission after examples from: Enterprise Integration Patterns book by Gregor Hohpe and Bobby Woolf, published by Addison-Wesley. This data sheet is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS SUMMARY.

Microsoft, Visual Studio, Windows, Windows Server and Windows Vista, are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. All other trademarks are property of their respective owners.